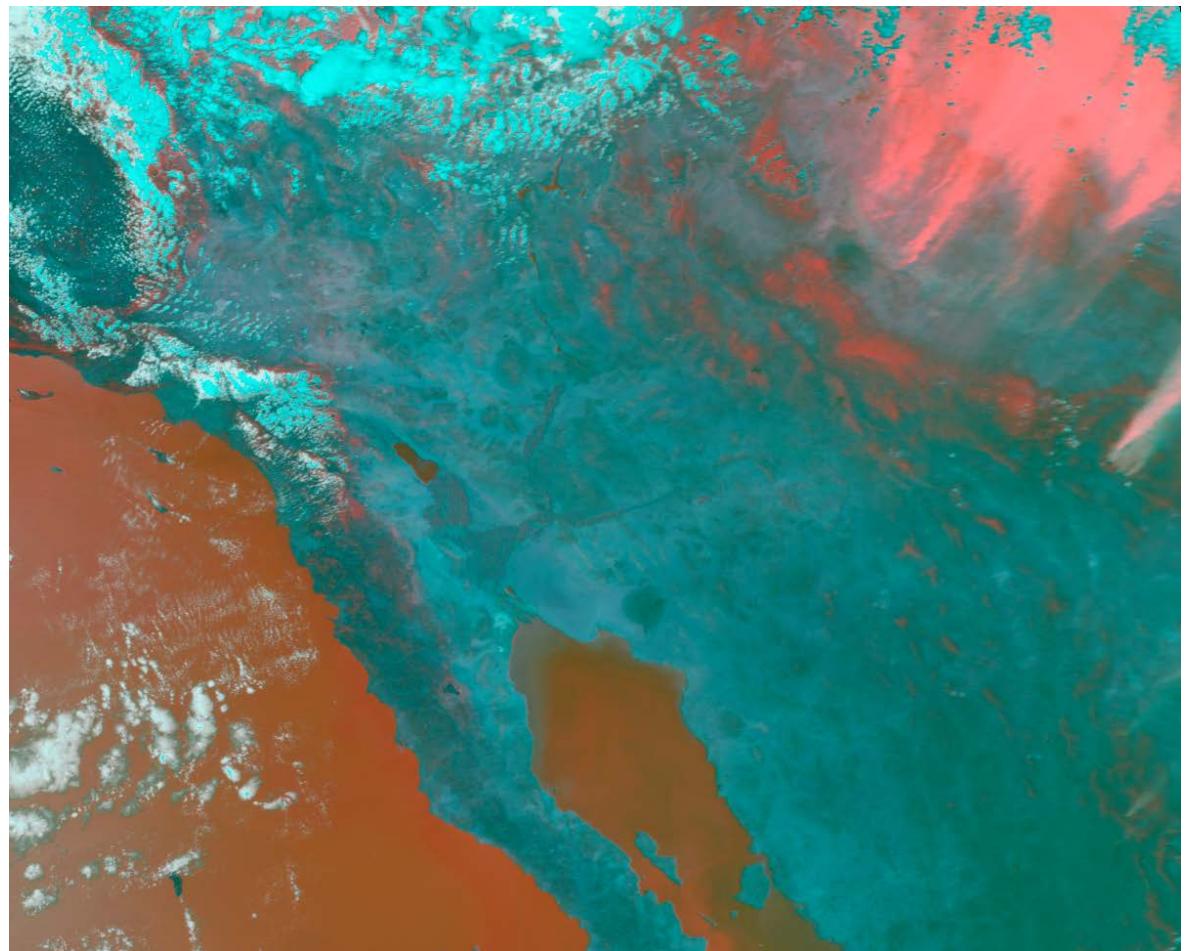Cold Regions Research and Engineering Laboratory

ERDC 6.2 Geospatial Research and Engineering (GRE) ARTEMIS STO-R DUST-CLOUD

Subjective Mapping of Dust-Emission Sources by Using MODIS Imagery

Reproducibility Assessment

Samantha N. Sinclair and Sandra L. Jones   May 2017

Subjective Mapping of Dust-Emission Sources by Using MODIS Imagery

Reproducibility Assessment

Samantha N. Sinclair and Sandra L. Jones

U.S. Army Engineer Research and Development Center (ERDC)
Cold Regions Research and Engineering Laboratory (CRREL)
72 Lyme Road
Hanover, NH 03755-1290

Final Report

Abstract

Dust storms (5 to 100 km across) often originate from multiple dust-emission sources (1 to 10 km across). Remote-sensing-based dust-source identification is a challenge. A previous study developed a subjective approach for mapping dust sources by using enhanced MODIS satellite imagery; therefore, this study conducted mapping exercises to assess the reproducibility of this technique amongst multiple analysts and in different regions. Multiple staff members independently analyzed satellite imagery for mappable dust sources for Southwest Asia and the Southwest United States. Mapped points were considered reproducible if the location of dust emission plumes identified by all participants could be constrained to a 10 km buffer. Using this definition, point-source locations were 28% reproducible in Southwest Asia and 85% reproducible in the Southwest United States. Increasing the allotted buffer to 15 km, however, improved results to 71% and 100%, respectively. Mapped dust sources for Southwest Asia were compared to geomorphic landform maps. At the 1:750,000 map scale, points mapped by all analysts for a single dust plume tended to overlay one landform, while at the 1:100,000 map scale, points were strewn across several landforms. Results suggest that the methodology is reproducible for certain applications but that location-uncertainty tolerance affects perceived conclusions.

Contents

Figures and Tables

Figures

Tables

Preface

Acronyms and Abbreviations

ARTEMIS STO-R	Army Terrestrial-Environmental Modeling and Intelligence System Science Technology Objective—Research
ASOS	Automated Surface Observing System
COAMPS	Coupled Ocean-Atmosphere Mesosphere Prediction System
CRREL	Cold Regions Research and Engineering Laboratory
DUST-CLOUD	Dynamic Undisturbed Soils Testbed to Characterize Local Origins and Uncertainties of Dust
ERDC	U.S. Army Engineer Research and Development Center
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
GIS	Geospatial Information System
GRE	Geospatial Research and Engineering
HDF	Hierarchical Data Format
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
TIFF	Tagged Image File Format
UTC	Coordinated Universal Time
VIIRS	Visible Infrared Imager Radiometer Suite
WGS	World Geodetic System

1   Introduction

1.1   Background

Atmospheric soil and mineral dust can significantly influence a variety of global-scale processes, such as the radiation budget of the atmosphere, biogeochemical reactions that occur in the ocean, and global climate (Shinn et al. 2000; Mahowald et al. 2005, 2014; Ravi et al. 2011; Webb et al. 2012; Huang et al. 2014; Knippertz and Stuut 2014; Skiles et al. 2015; Wang et al. 2017). On regional and local scales, lofted dust can negatively affect visibility, mobility, communication, and human health (Rushing et al. 2005; Rushing and Tingle 2006; De Longueville et al. 2010; Okin et al. 2011; Al-Hemoud et al. 2017; Middleton 2017;). As a result, understanding the processes that control spatial and temporal patterns of atmospheric dust occurrence has become a priority for the research, military, operational forecasting, and hazard mitigation communities (e.g., Knippertz and Stuut 2014; Sprigg et al. 2014; Shepherd et al. 2016).

Accurate dust-source characterizations are critical for effectively modeling dust storms and their associated hazards. Numerous remote-sensing studies have attempted to identify dust-emission-source locations (e.g., Legrand et al. 1994; Torres et al. 1998; Prospero et al. 2002; Miller 2003; Bullard et al. 2008; Baddock et al. 2009; Walker et al. 2009; Ginoux et al. 2012; Jafari and Malekian 2015; Moridnejad et al. 2015; Zhang et al., forthcoming). Historically, satellite-based approaches have struggled with dust-plume origin detection. This is because synoptic and mesoscale dust storms on the order of 5 to 100 km rarely originate from a single source but rather from the amalgamation of several point-source dust plumes. These smaller plumes stem from multiple point sources (i.e., plume heads 1 to 10 km across) that can evade coarser-resolution satellite detection.

Advances in satellite image resolution (1 km or better) have greatly enhanced the ability to detect plume heads remotely (Walker et al. 2009; Muhs et al. 2014). Walker et al. (2009) developed an approach to manually map plume-head point sources in a geospatial information system (GIS) framework using Moderate Resolution Imaging Spectroradiometer (MODIS) imagery processed through a dust-enhancement algorithm. With this technique, a location is digitized and archived if the analyst observes an unobscured plume head in the imagery. Although analysts are able to

map dust sources at the satellite-image grid scale, location analysis error is almost unavoidable. Airborne dust must be sufficiently elevated for overland dust enhancement algorithms to work; otherwise, thermal and reflectance contrast signals between the dust and the underlying surface may not be discernable. By the time the dust has reached a sufficient altitude, it is likely to have traveled downstream from its deflating source. As such, errors in digitized source location may be on the order of 1 to 10 km (Walker et al. 2009).

Even with this inherent uncertainty, the Walker et al. (2009) technique is a popular approach for dust-source identification and is often used as a proxy for dust-source observations (e.g., Lary et al. 2016). The Walker et al. (2009) methodology is also currently being used to incorporate dust sources into the Navy's Coupled Ocean-Atmosphere Mesosphere Prediction System (COAMPS; e.g., Liu et al. 2007; Walker et al. 2009), which to date remains the highest-resolution dust-source treatment in an operational military weather forecast model. To our knowledge, however, the sensitivity of the Walker et al. (2009) methodology to analyst subjectivity has never been formally assessed.

1.2   Objective

The goal of this study is to examine the role of analyst subjectivity on the reproducibility of the Walker et al. (2009) technique.

1.3   Approach

A group of analysts independently mapped plume-head point sources for two predetermined case-study dust events, one in the Southwest United States and one in Southwest Asia. Section 2 of this report provides step-by-step instructions for an adapted version of the Walker et al. (2009) methodology and outlines the procedures and criteria used to determine if dust was present in a satellite image. It also includes the GIS mapping procedures. Section 3 outlines the two geographical regions and provides background information pertinent to each case study. Section 4 reviews the resultant plume-head point-source maps. Section 5 discusses final reproducibility assessments and limitations of the approach, and Section 6 identifies potential areas of future work.

2 Methods

For each case-study region, a single analyst downloaded Automated Surface Observing System (ASOS) weather data to identify dusty days reported in local weather station records. The same analyst then downloaded and processed true-color and dust-enhanced MODIS satellite imagery associated with each day of interest and selected a single time period for each regional case study based on a perceived potential for numerous mappable plume-head point sources. After reviewing mapping procedures and techniques, a group of additional analysts then individually mapped plume-head point sources in the images selected for each region by using ArcMap 10.3 and assigned a quality score (confidence level) to each mapped point. Results from each analyst were subsequently aggregated together and assessed for reproducibility.

Because of a general lack of in situ dust-emission observations, there is no way to definitively corroborate the precise locations of historically emitting dust sources on a regional scale or to verify that point sources subjectively identified by an analyst are actually real sources. Thus, quantitative assessment is limited in nature to point-placement reproducibility amongst analysts. In other words, this report addresses the sensitivity of plume-head location placement to analyst interpretation. It does not attempt to quantify the ability of individual group members to reproduce actual emission-source patterns. As a result, the reproducibility assessments in this report are based only on the plume heads that were identified by all participating analysts. Of those points, a mapped plume-head point source is considered reproducible if all point-source markers are constrained to a 10 km buffer, a limit defined by the potential for advection error when using the Walker et al. (2009) methodology.

In addition to mapping plume-head point sources, an overlay analysis using geomorphic landform maps was also conducted for the Southwest Asia case study. Certain arid-region landform types tend to be richer in dust-sized material than others (e.g., Bullard et al. 2011; Sweeney et al. 2011, 2016; Sweeney and Mason 2013) because processes that dictate landform evolution also influence soil development. As a result, some researchers believe that landform designations can be used as a suitable proxy for soil attributes like dust-emission potential (e.g., Bullard et al. 2008; Bacon and McDonald 2016). For example, dry lakebeds filled with unconsolidated

fine-grained sediments are commonly considered a significant dust-emission source globally (e.g., Prospero et al. 2002). Because of this important link between landform type and dust emissions, a more comprehensive understanding of the geomorphic controls that dictate atmospheric dust is necessary to enhance weather and climate models (Baddock et al. 2016). Thus, an overlay analysis may offer insight as to whether the Walker et al. (2009) technique may be a reliable option for validating various geomorphic-based dust-source characterization approaches.

2.1 Satellite imagery acquisition and selection process

All satellite images used in this study were derived from MODIS data, which have been collected by the Earth Observing System Terra and Aqua satellites since early 2000 and mid-2002, respectively (see https://modis.gsfc.nasa.gov/). ASOS weather station data ranging from 1 January 2000 to 30 June 2016 was downloaded for the case-study regions and inspected for accounts of airborne dust to help select appropriate satellite imagery for the reproducibility experiments. These ASOS stations record sub-hourly meteorological data and are found primarily at airports across the globe (Nadolski 1998). Appendix A provides step-by-step instructions for ASOS data acquisition.

Periods of ASOS station records with entries of "blowing dust," "dust devils," or "sandstorm" in the obscuration-to-vision column (i.e., the "obscure" column) were noted for consideration. These data, however, required further reduction as not all dust events identified via ground-based sensors are detectable by satellites as previously discussed. Visibility observations greater than approximately 32 km were used to filter out isolated events or dust events likely undetectable by the MODIS sensors. Viable case-study options were further reduced to periods with several (five to ten) weather stations simultaneously reporting dust or stations reporting long periods (hours to days) of near continuous blowing dust, as these are strong indicators of widespread dust lofting through synoptic or mesoscale forcing mechanisms.

For each potential period identified via the ASOS record inspection, a single analyst downloaded MODIS Level 1B 1 km Calibrated Radiance granules. These granules were originally downloaded as HDF (hierarchical data format) files, post-processed, and reformatted into georeferenced TIFF (tagged image file format) imagery that could be directly imported into ArcGIS. Three unique images were then generated from each acquired

granule (e.g., Figure 1) using the channels listed in Table 1 and a script developed in Python (see Appendix B). The first is a true-color image, a bird's eye representation of the Earth from space without the addition of any filtering or post processing (e.g., Figure 1a). The other two images were created using false-color dust-enhancement algorithms (e.g., Figures 1b and 1c).

The first dust-enhanced image is rendered using a technique by Miller (2003), which uses visible, near-infrared, thermal-infrared, and water-vapor channels to distinguish elevated dust from the underlying background. The Miller (2003) approach uses different equations to set the resulting red color over land and water. Lofted dust appears pink, landscapes have blue and green hues, water and steep terrain are red, and clouds appear aqua or cyan (see Figure 1b). For a detailed overview of the algorithm procedure, see section 3 of Miller (2003).

The second dust-enhanced image is generated using an algorithm originally developed by the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) for use with their geostationary satellite data. This algorithm has since been adapted for other satellite platforms (e.g., Lensky and Rosenfeld 2008; Brindley et al. 2012). Unlike the Miller (2003) approach, the EUMETSAT technique does not require use of any visible channels, making it useful for both day and nighttime dust detection. Lofted dust often appears hot pink against a purple landscape below, and thick clouds typically appear red, which can camouflage the pink dust in some instances (see Figure 1c). An overview of the EUMETSAT algorithm and procedures for setting imagery color palettes are available in section 2 of Brindley et al. (2012), and step-by-step instructions to render MODIS imagery are located at the end of this report in Appendix B. Imagery created with true-color settings, the Miller (2003) algorithm, and the EUMETSAT algorithm are labeled "True Color," "MILLER," and "EUMETSET," respectively, throughout this report.

Figure 1. MODIS satellite imagery of the Southwest United States and Mexico. *Panels A*, *B*, and *C* depict the area outlined by the red box within the inset of *Panel A*. *Panel A* is a True Color image that depicts *white* clouds against a *brown* landscape; a bird's eye representation of Earth. *Panel B* is a MILLER image that depicts *aqua* colored clouds against a *dark blue/green* landscape. Lofted dust appears *pink* while high-relief topography appears *red*. The *white dashed box* depicts a region that does not contain any lofted dust. *Panel C* is a EUMETSAT image that depicts *red* clouds against a *purple* landscape. Lofted dust appears *hot pink*. The *white circle* in each panel depicts a dust event. Note the *pink* dust that can be seen in *Panels B* and *C* is undetectable in *Panel A*.
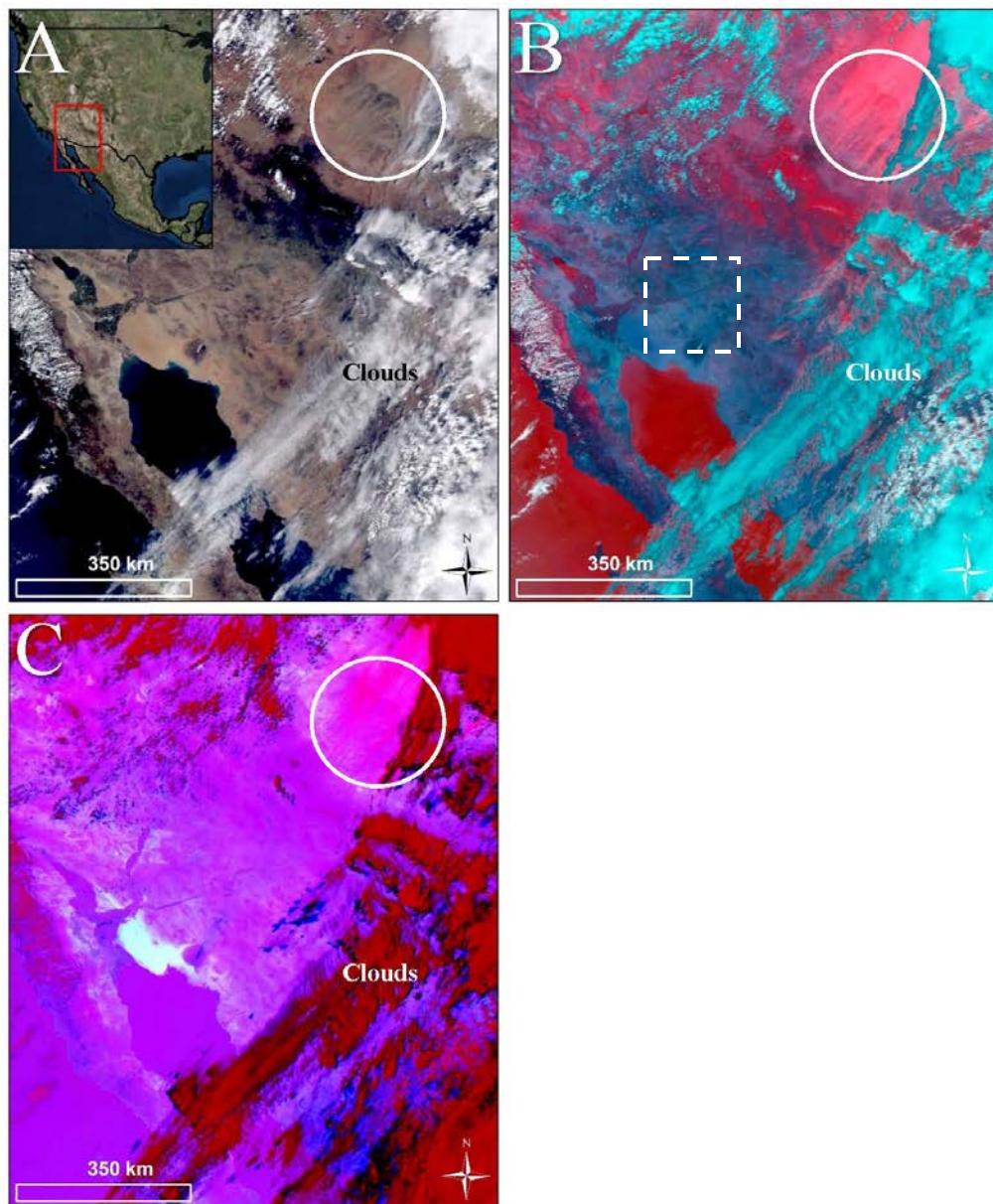
Table 1. MODIS channels used to generate case-study imagery.

MODIS Channel	Description	Band Width (μm)
1	Red	0.620–0.670
2	Near Infrared	0.841–0.876
3	Blue	0.459–0.479
4	Green	0.545–0.565
26	Water Vapor	1.360–1.390
29	Water Vapor	8.400–8.700
31	Infrared	10.780–11.280
32	Infrared	11.770–12.270

Of all the MODIS images that were downloaded for this study, only a select few were useful for mapping plume-head point sources. This was because of a variety of reasons, including but not limited to the Terra and Aqua satellites not passing over the area of interest during the time of the dust storm; the presence of cloud coverage obscuring the dust below; or the occurrence of a low-concentration, short-lived, insufficiently lofted, or nighttime dust storm. In addition, analysts were not able to map lofted dust in all imagery that coincided with an event as, in many instances, dust appeared as an effusive cloud without discernable point sources that could identified or mapped (Figure 2).

Figure 2. True Color (*left*) and MILLER (*right*) images depicting a dust event on the Colorado Plateau in northeast Arizona in the area outlined by the red box. Note that while there is a considerable amount of pink-colored portions of the MILLER image, discernable plume-head point sources are not apparent and therefore cannot be deduced or mapped.
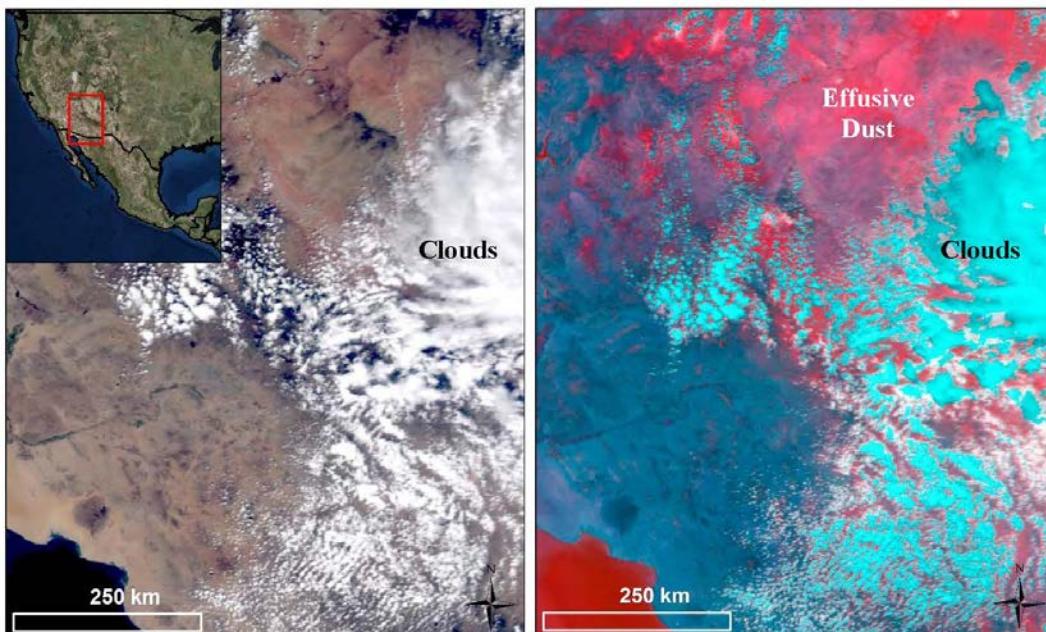
Other images contained plume-head-like features that initially looked like mappable sources; but further inspection identified them as thin clouds, forest fires, or high-relief ridges. Clouds often looked deceivingly like lofted dust in the MILLER images, particularly when relatively thin (e.g., Figure 3). To ensure that the team did not mistakenly map clouds as dust, analysts used the True Color and EUMETSAT images to clearly define the location of any clouds in the region before mapping plume-head point sources. Large-scale forest fires and certain landforms also portrayed dust-like characteristics in the MILLER images. Some high-relief landforms, such as narrow mountain ridges, appeared reddish/pink in the images, mimicking the appearance of a long dust-plume tail (see Figure 3). Smoke from forest fires appeared as a plume-like feature that emanated from a single point source and looked nearly identical to plume heads in the MIL-LER and EUMETSAT images (Figure 4). To ensure that these features were not accidentally mapped as dust sources, the analysts used the True Color image to evaluate where the dust sources were coming from off the landscape. Narrow mountain ridges were easily identifiable in the True Color images and could immediately be ruled out, and smoke typically originated in forested terrains that were very unlikely to be large dust pro-ducers.

2.2   Mapping dust-plume-head point sources

Once a single set of the True Color and dust-enhanced images was selected from the remaining list of options for each case-study region, the analysts mapped plume heads using ArcMap 10.3 software. This involved upload-ing the processed MODIS imagery and creating and editing shapefiles of point markers for each region. All participating analysts then inde-pendently mapped point sources they identified through subjective im-agery interpretation. Each analyst also assigned a quality score to every mapped point as a means to communicate user confidence in plume-head mapping decisions. Quality scores were ranked on a qualitative scale from 1 to 3. Users assigned a particular point a quality score of 3 only when they were confident in the placement of the point and felt certain dust was orig-inating from that location. A quality score of 2 meant users were almost certain dust was coming from that location but were not entirely confi-dent. A quality score of 1 meant the user was not certain in the placement of the point. Appendix C outlines the detailed step-by-step instructions for the mapping procedures and quality score assignment followed in ArcMap.

Figure 3. True Color (*top left*), MILLER (*top right*), and EUMETSAT (*bottom left*) images for the region outlined by the *red box* over portions of the Southwest United States and Mexico. These images highlight the potential for high-relief terrain features to be misinterpreted as dust. Note the *light pink* colored feature to the east of the Gulf of California in the MILLER image that resembles the appearance of lofted dust. When compared to the True Color image, it is apparent that these features are clouds. The *white circle* in the MILLER image highlights high-relief terrain that has the appearance of highly concentrated lofted dust. This same region outlined in the EUMETSAT image, however, does not show any indication of lofted dust. The True Color image reveals this region is composed of mountainous terrain.
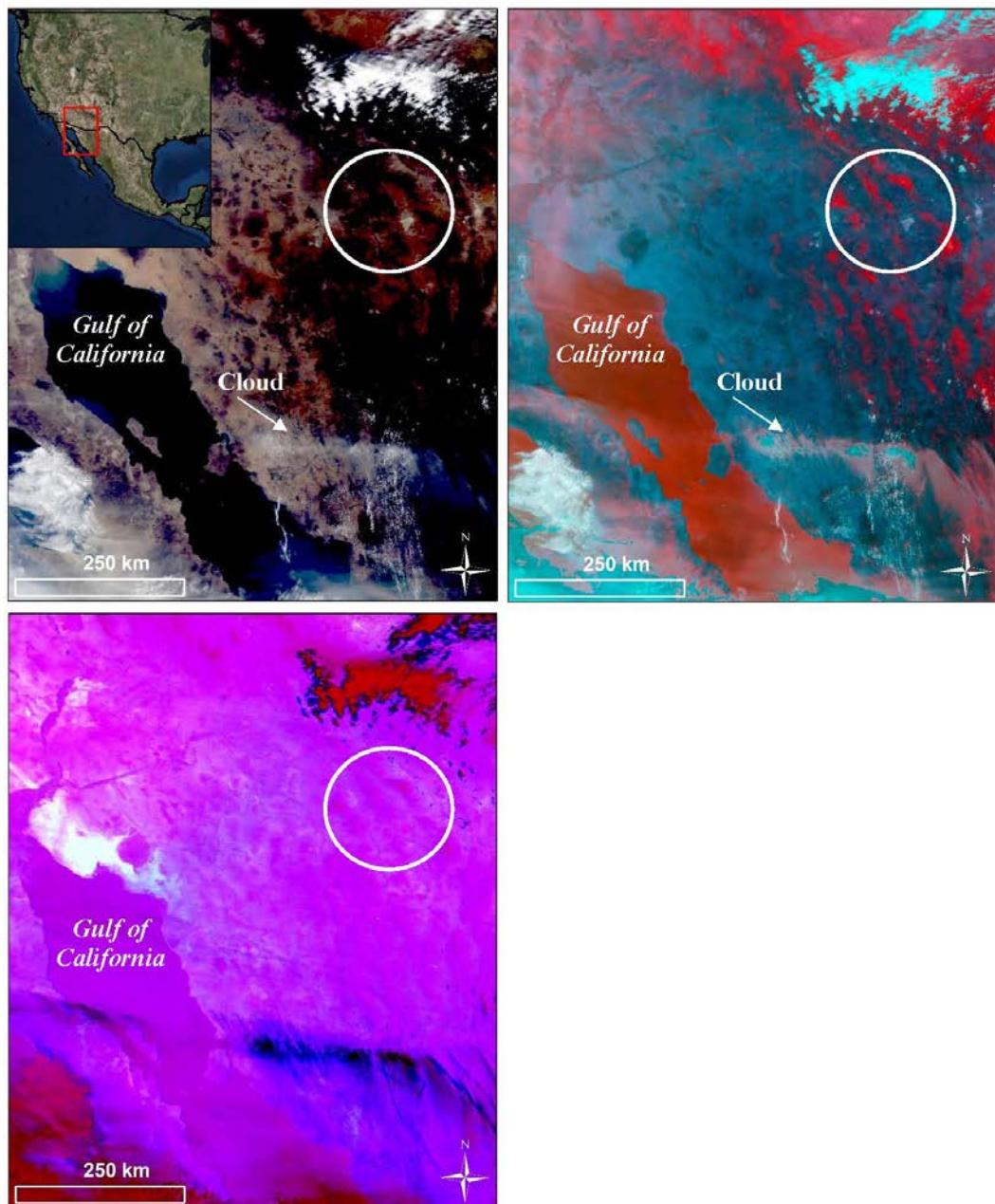
Figure 4. As in Fig. 3 but for smoke. True Color (*top left*), MILLER (*top right*), and EUMETSAT (*bottom left*) images for the region outlined by the *red box* over portions of the Southwest United States and Mexico. The MILLAR and EUMETSAT images reveal a mesoscale dust event in the northeast corner of the domain. A single plume located south of the dust storm also looks as if it could be lofted dust. This same feature is apparent in the True Color image but very clearly originates from forested terrain, indicating that it is very likely smoke from a forest fire.
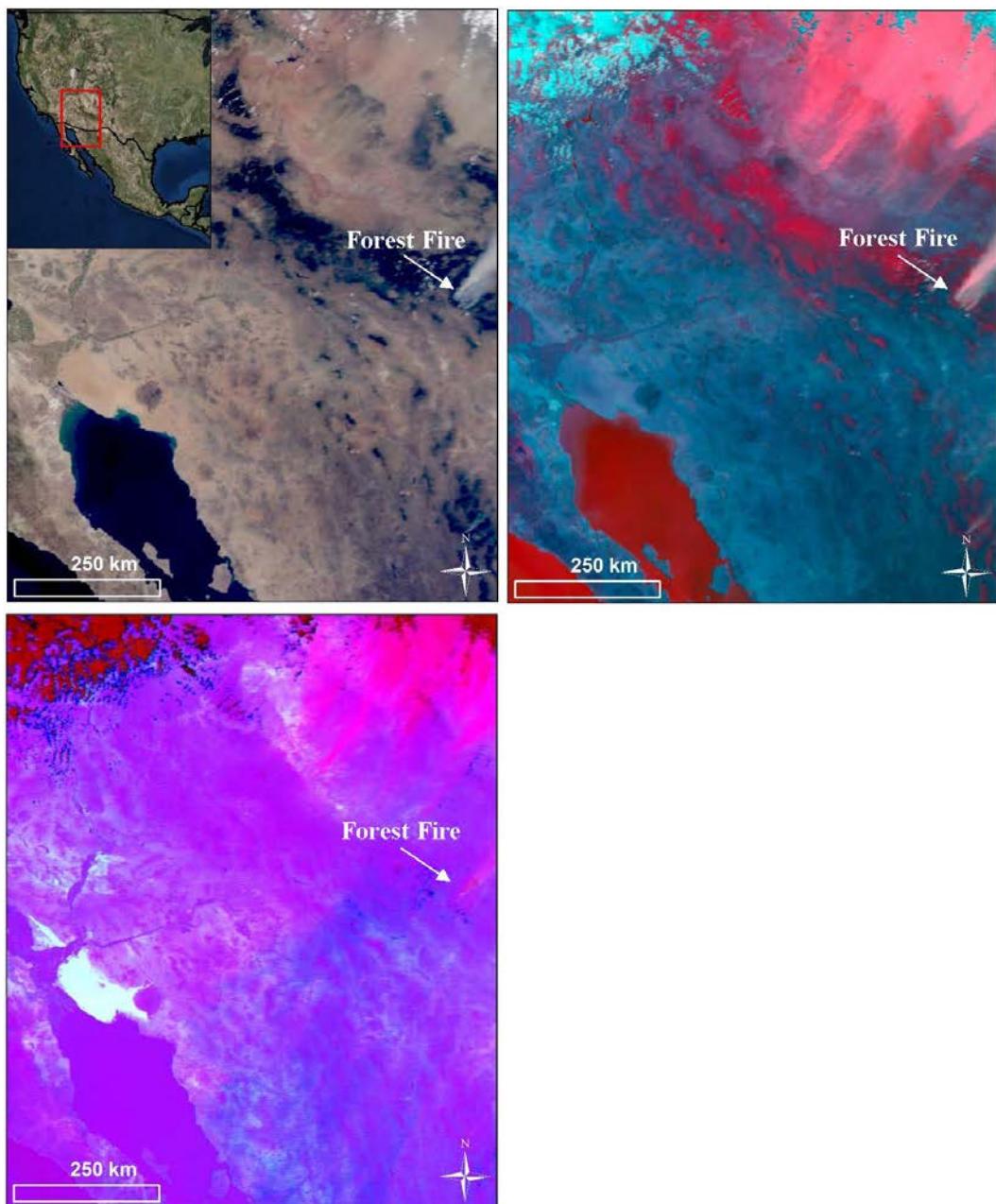
3 Case-Study Descriptions

3.1 Southwest Asia

The desert region of Southwest Asia is composed of several countries, including Saudi Arabia, Oman, the United Arab Emirates, Iran, Afghanistan, Pakistan, and India (Figure 5). The geomorphology of this region is highly variable and includes mountainous terrain, river valleys, plateaus, dry washes, ephemeral lakes, alluvial fans, sand dunes, and salt plains (Goudie and Middleton 2006; Affleck et al. 2011). Vegetation is sparse but diverse due to the variety of unique landscapes present. Common types of vegetation include camel brush, broom grass, buckthorn, and a variety of trees in the mountainous regions (Khaurin 2003; Breckle 2007).

Figure 5. Map of Southwest Asia (as outlined by the *red box*) depicting western India, Pakistan, Iran, Afghanistan, Oman, the United Arab Emirates, and eastern Saudi Arabia. The *white box* shows the location of the Hamun dry lakes, which are an important dust source in the region. The enlarged image box in the *top right* depicts the individual lakebed boundaries (modified from Rashki et al. 2013). Dust is lofted from the Hamun region and transported in a southeasterly direction around the Chagai Hills and into Pakistan.
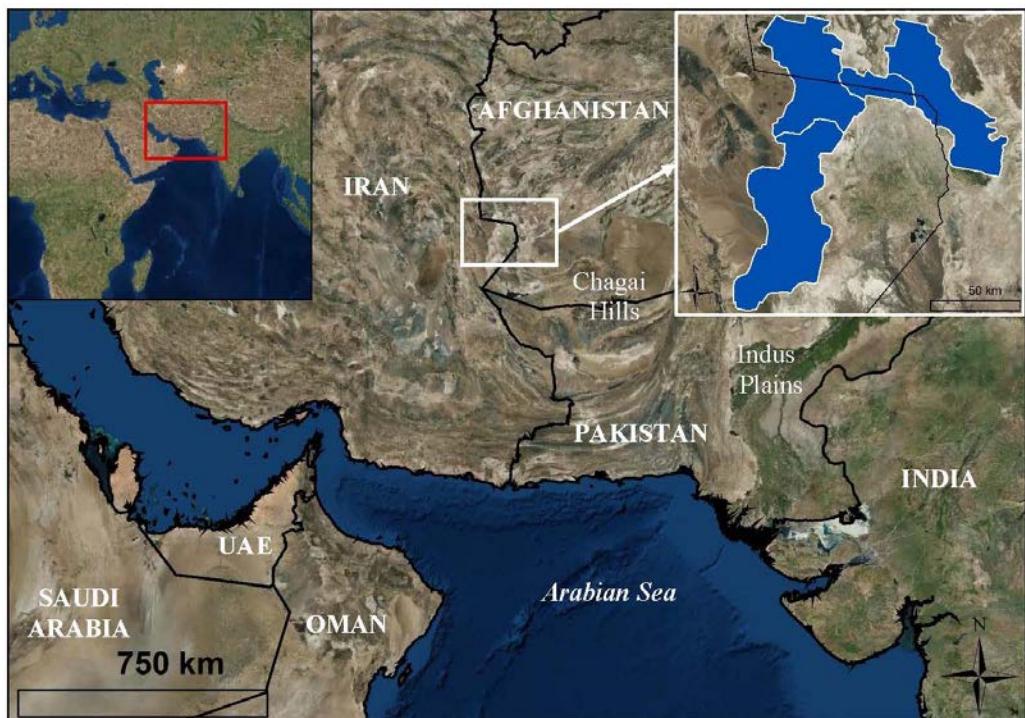
The majority of dust storms in Southwest Asia occur at the convergence between the borders of Iran, Pakistan, and Afghanistan (Goudie and Middleton 2006). Other sources of dust in the region include the coast of the

Arabian Sea in Iran and the Indus Plains in Pakistan, and dust originating in the Sahara Desert is also frequently transported to the area (see Figure 5) (Goudie and Middleton 2006). At the convergence between Iran, Pakistan, and Afghanistan, dust sources are located in the lowland valleys between mountain peaks. Major dust emitters in the region include alluvial fans and ephemeral lakes, such as the Hamun dry lakes and the deltaic fan associated with the Helmand River. Dense dust plumes often originate from these sources and are transported by strong northerly winds blowing to the southeast through lowland areas and around the Chagai Hills (see Figure 5) (Goudie and Middleton 2006; Muhs et al. 2014). This is a dominant pattern often seen in the area and is representative of the specific dust storm that took place on 16 September 2011 and was mapped in this study. Images were generated from the MODIS Terra 0650 UTC[*] (1120 local time in Southwest Asia) granule.

3.2   Southwest United States

The desert region of the Southwest United States encompasses portions of California, Nevada, Utah, Arizona, New Mexico, and Texas between the Pacific Ocean and the Rocky Mountains and is composed of four major deserts: the Great Basin, Mojave, Sonoran, and Chihuahuan (Figure 6). The regional geomorphology includes wash plains, dry lakebeds, playas, sand dunes, floodplains, alluvial plains, and isolated mountain ranges shaped by erosional processes. Vegetation in the region, though sparse, does not conform to the near-void vegetation patterns typically seen in other desert regions globally. Instead, a denser population of desert shrub, brush, grasses, and cacti, including plants such as cheat grass, creosote, mesquite, atriplex, and ironwood, cover a larger percentage of the surface.

Dust storms occur throughout the year in the Southwest United States (driven by prevailing wind patterns), though seasonal fluctuations in the Pacific High (a semi-permanent, subtropical anticyclone located over the northeastern Pacific Ocean) and the Jet Stream drive variations in the number and intensity of dust storms observed throughout the year (Adams and Comrie 1997; Tong et al. 2012). During the monsoon season (July to mid-September), dust events are usually induced by localized thunderstorms. The dust storm mapped in this study is associated with a weather front that moved across the Colorado Plateau in Arizona on 2 April 2003.

---

[*] Coordinated Universal Time

Images were generated from the MODIS Terra 2050 UTC (1250 local time in the Southwest United States) granule.

Figure 6. Map of the Southwest United States desert regions. The boundaries of the four major deserts in this region, including the Great Basin, Mojave Desert, Sonoran Desert, and Chihuahuan Desert, are outlined (modified from EPA 2012).

4 Results

4.1 Southwest Asia

Table 2, Figure 7, and Figure 8 show the results from the reproducibility experiment conducted for Southwest Asia. As shown in Table 2, the number of plume-head point sources mapped by each analyst ranged from 10 to 28, with an average of 19 ± 6.9. Mean quality scores between analysts ranged from 1.94 to 2.5, with an average of 2.15 ± 0.22. The analyst that mapped the greatest number of point sources was second to least confident in their decisions while the analyst that mapped the fewest points was the most confident. In total, there were seven plume heads that all five analysts mapped (see Figures 7 and 8). Of these, two plume heads fell within a 10 km buffer while all others fell within a 15 km (three plume heads), 25 km (one plume head), or 35 km (one plume head) buffer. Other plume heads depicted in the image were only mapped by a portion (four or fewer analysts) of the team and thus were not considered for reproducibility evaluation.

Results were mapped onto a 1:100,000 and a 1:750,000 geomorphic landform map produced by the Desert Research Institute (see McDonald et al. 2013 for landform map generation and attribute details). At a 1:750,000 map scale, plume-head point sources mapped by all five analysts for a single plume head fell within the same type of landform (alluvial plain), while at the 1:100,000 map scale, the same points were strewn across several different landform types (delta plain, playa, wet playa, and sand sheets; see Figure 7).

Table 2. Number of mapped dust sources and mean quality scores for the Southwest Asia 16 September 2011 0650 UTC case study.

Analyst Identification Number	Number of Mapped Dust Sources	Mean Quality Score
1	28	1.96
2	14	2.29
3	17	1.94
4	10	2.5
5	26	2.04

Figure 7. MILLER image (*top*) depicting a dust event on 16 September 2011 in Southwest Asia and associated dust-plume-head markers mapped by five analysts. The region in the *white rectangle* is enlarged in the *bottom* panels. The 1:750,000 (*bottom left*) and 1:100,000 (*bottom right*) scale geomorphic landform maps are overlain with mapped plume-head point sources. The *oval* in each panel outlines five points placed by each analyst for single plume head.
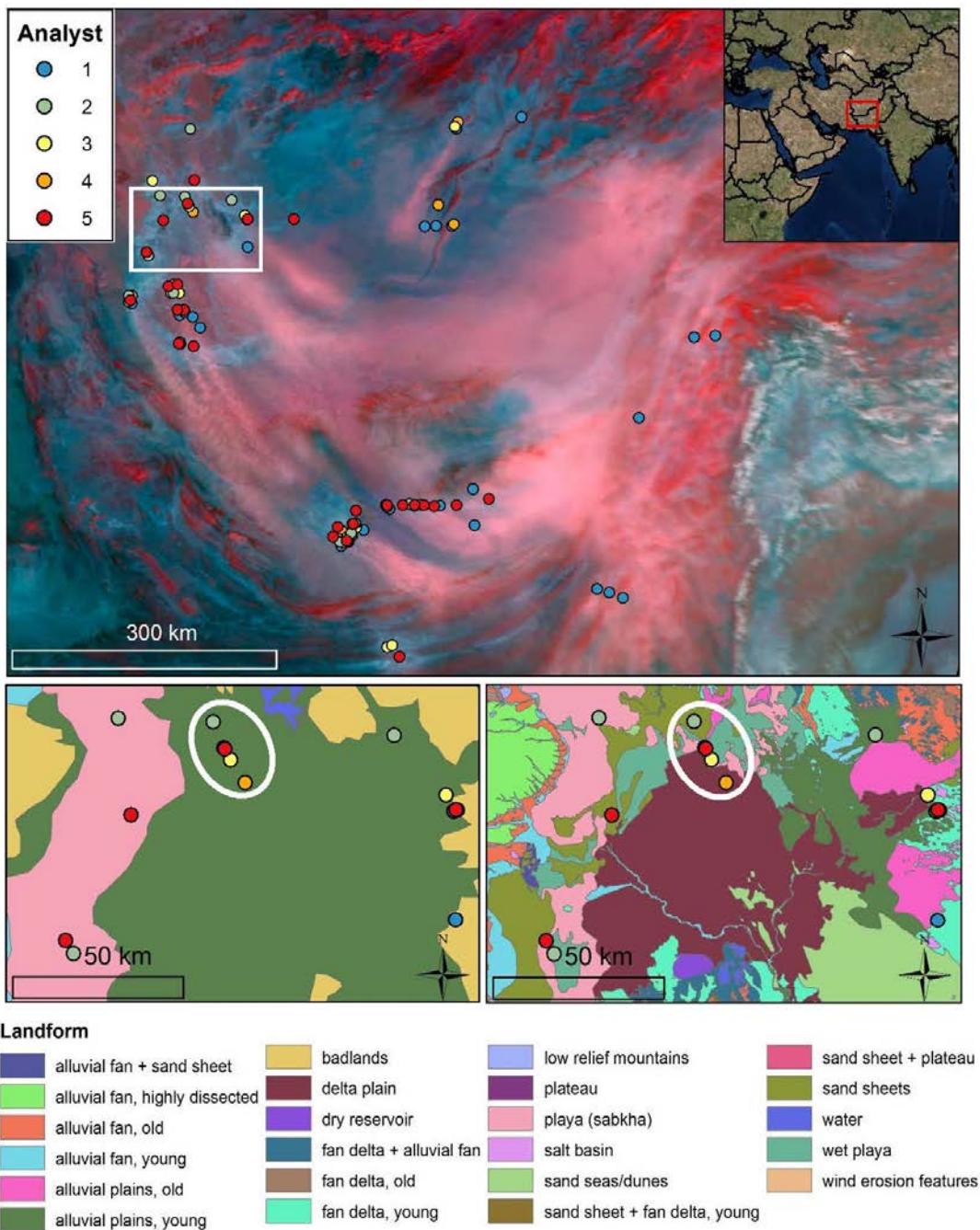
Figure 8. Quality scores assigned by each of the five analysts to points mapped in the Southwest Asia case study. Points marked with high confidence (*3*) are *green*, moderate confidence (*2*) are *yellow*, and low confidence (*1*) are *red*. Each panel depicts the points placed by an individual analyst. Note that points mapped by all analysts were assigned a higher quality score (and were therefore easier to identify and map) than those mapped by only one or two analysts.
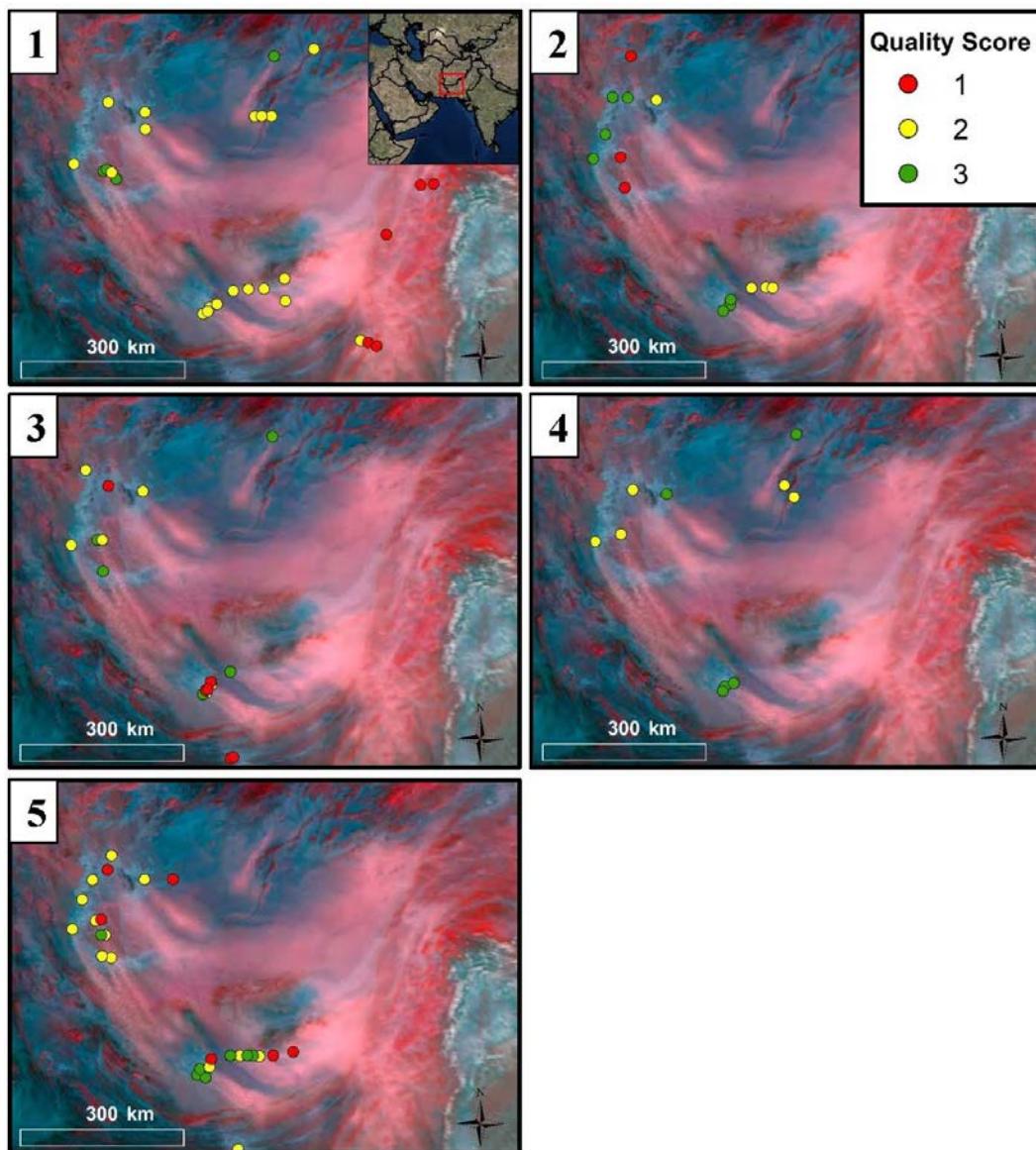
4.2   Southwest United States

Table 3, Figure 9, and Figure 10 show results from the second reproduci-bility experiment conducted for the Southwest United States. One of the team analysts could not participate in the second experiment, so only four of the five initial analysts mapped this region. As shown in Table 3, the number of plume-head point sources mapped by each analyst ranged from 13 to 29, with an average of 22 ± 6.7. Mean quality scores between analysts ranged from 1.23 to 2.14, with an average of 1.77 ± 0.41. In total, there were seven plume heads that all four analysts mapped. Mapped sources for six of these plume heads fell within a 10 km buffer, while the other fell within a 25 km buffer. Of the six plume heads mapped within a 10 km buffer, three were constrained to a 5 km buffer and one to a 3 km buffer. Several points were mapped in regions that were not dust producing areas. One analyst mapped several plume-head like features that were actually thin clouds in the region (see Figure 9). Fortunately, that analyst marked all of those points with a quality score of 1, indicating a low level of confi-dence (see Figure 10). Landform maps for this region were not accessible at the time of the experiment, and therefore analysts did not conduct over-lay analyses.

Table 3. Number of mapped dust sources and mean quality scores for the Southwest United States 2 December 2003 2050 UTC case study.

Analyst Identification Number	Number of Mapped Dust Sources	Mean Quality Score
1	29	1.66
2	13	1.23
3	22	2.14
4	24	2.04

Figure 9. MILLER image (*left*) depicting a dust event on 2 April 2003 in the Southwest United States and the associated dust-plume head markers mapped by four analysts. Points mapped by each analyst are represented by different *colored circles*. Note the five points mapped by analyst 1 in the southeast corner of the image that are incorrectly placed on clouds. The area outlined in the *black rectangle* is enlarged on the *right*: a close-up of individual plume-head point sources mapped by each analyst. Also, note the variability in the number of plume heads subjectively identified by each analyst. Buffers placed around plume heads mapped by all analysts show a high degree of duplicability and are constrained to 3 km (*blue circle*), 5 km (*green circles*), and 10 km (*yellow circles*) buffers. This image does not include the 25 km buffer.
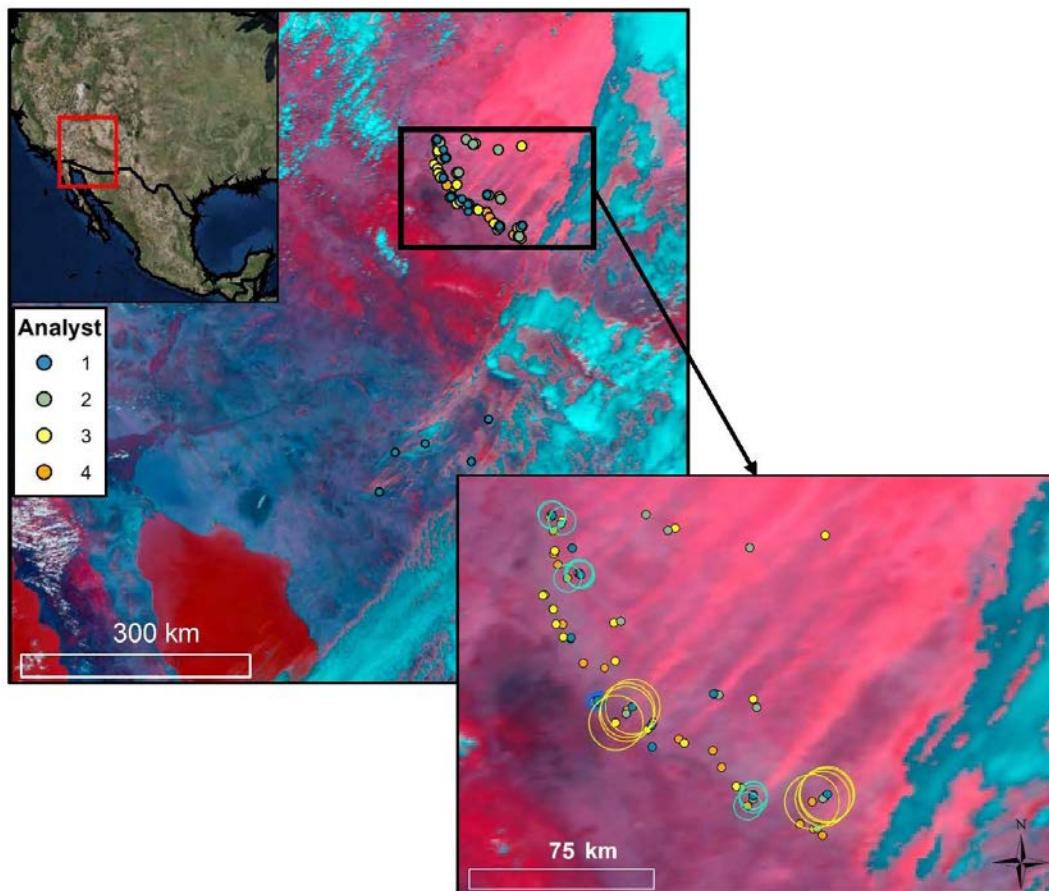
Figure 10. Quality scores assigned by each of the four analysts to points mapped in the Southwest United States case study (shown in Fig. 9). Points marked with high confidence (*3*) are *green*, moderate confidence (*2*) are *yellow*, and low confidence (*1*) are *red*. Each panel depicts the points placed by an individual analyst. As with the Southwest Asia study, plume heads marked by all analysts received a higher quality score than those mapped by just one or two analysts.
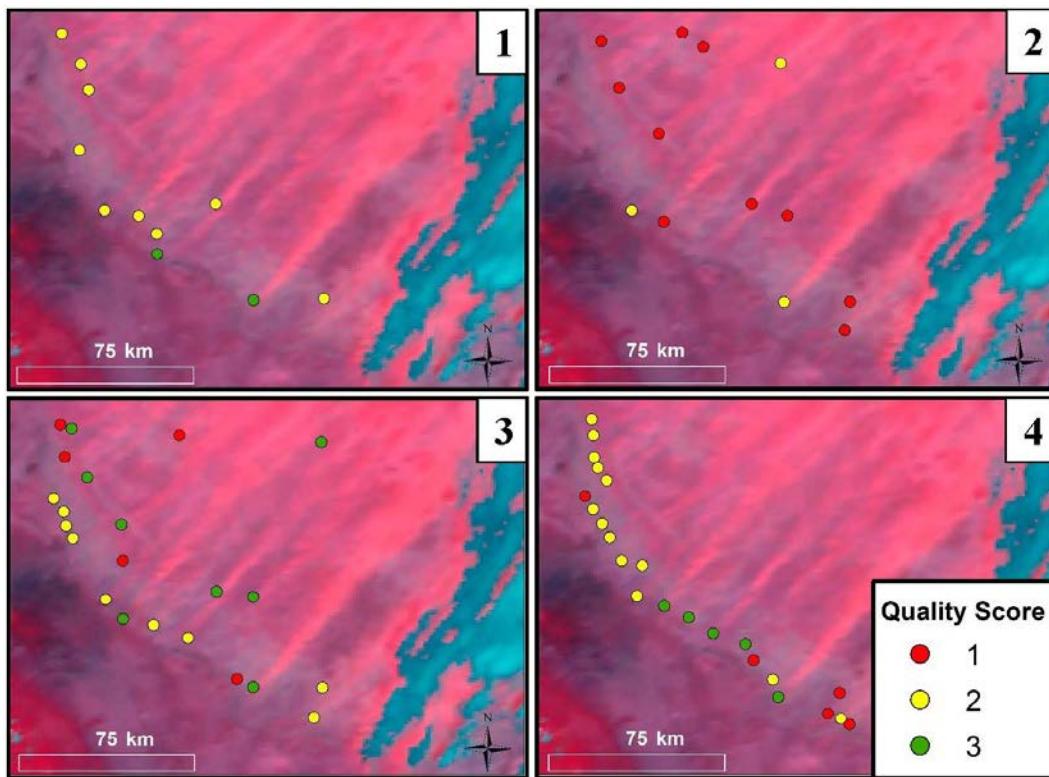
5   Discussion

Following the 10 km location uncertainty assumption per Walker et al. (2009), this reproducibility study considers points to be reproducible if markers submitted by each analyst fall within a 10 km buffer. In the Southwest Asia case study, only two of the seven plume-head point sources mapped by all participating analysts (~29%) could be constrained to a 10 km buffer. From the visual inspection of Figure 7 and from group discussions, however, the team of analysts determined that there are likely more than seven mappable plume-head point sources in the Southwest Asia imagery. In particular, the team agreed in a later group debate that four plume-head point sources that a fraction of the team (three or four analysts) mapped were indeed mappable sources. This result highlights the subjective nature of identifying and mapping plume-head point sources and suggests that analysts do not always define mappable sources in the same manner. It could, therefore, be beneficial to have a team of analysts, rather than a single user, apply this methodology to a given region as point sources overlooked by some team members may be identified and mapped by others.

In the Southwest United States study, six of the seven plume-head point sources mapped by all participating analysts (~86%) could be constrained to a 10 km buffer. As with the Southwest Asia experiment, group discussions led to an agreement that there are likely more than seven mappable plume-head point sources depicted in the Southwest United States imagery. A thorough group discussion deemed that three plume-head point sources mapped by a fraction of the team (two or three analysts) were mappable sources, though again, there is no way to verify the accuracy of the group consensus.

Our study supports the use of the Walker et al. (2009) methodology for global dust-emission-activity characterization as the analysts were able to successfully map two markedly different, regional-scale desert environments. The results from both the Southwest Asia and U.S. case studies, however, highlight that although the Walker et al. (2009) methodology is reproducible to an extent, it may not produce one-to-one results between multiple analysts. Our findings also demonstrate that dust-source characterization products created via the Walker et al. (2009) methodology may not be suitable proxies for observational "truth." If the definition of reproducible was expanded to include point sources mapped to within a 15 km

buffer, however, five of the seven (~71%) Southwest Asia points and all seven (100%) of the Southwest United States points would be considered reproducible. Thus, aspects of the final product's format and intended application largely affect conclusions about the reproducibility of the Walker et al. (2009) technique. For example, Walker et al. (2009) applied their final dataset to a dust transport model as a preferential dust-source region map interpolated to a 27 km grid. This grid spacing resolution is well outside the bounds of expected detection and analyst-induced location-error uncertainty.

Of the 183 total points mapped by the analysts in this study, only five were deemed to be incorrectly placed at definitively non-dust-emission sources (see Figure 9). This demonstrates that users encountered minimal difficulty in differentiating between plume-head point-source locations and false-positive "plume-head lookalikes," like forest fires or high-relief terrain. The five points that were incorrectly mapped as plume-head point sources (and were in fact clouds) were assigned a quality score of 1, which indicates that the quality score is an important aspect of this methodology that can be used to winnow out unusable data. Plume-head point sources mapped by one or two analysts typically received a quality score of 1, while 89% (in Southwest Asia) and 79% (in the Southwest United States) of plume-head point sources mapped by all analysts received a quality score of 2 or 3 (see Figures 8 and 10). User confidence could therefore be used as a predictive measure to identify mapped points that are likely actual dust-emission sources.

Our geomorphic overlay assessment suggests that the 1 km resolution of MODIS imagery coupled with the mapped point-source location uncertainty may prohibit use of the Walker et al. (2009) technique for linking dust-emission sources to a particular landform type or geographic location outside of very coarse-resolution scale analyses. Previous studies have shown that individual geomorphic landforms may not always act as homogeneous dust-emission sources and may consist of both erodible and unerodible portions (Walker et al. 2009; King et al. 2011; Sweeney et al. 2011). Additionally, some dust-emission sources may be associated with individual fields or farmland (e.g., Gillette 1999). Dust plumes emitted from these relatively small, localized sources are likely too narrow for MODIS sensors to detect.

6   Opportunities for Future Research

The inability of the MODIS data to display narrow dust plumes is a limiting factor when characterizing dust-emission sources for highly variable landscapes. Newer and higher-resolution multispectral satellite capabilities, such as the Visible Infrared Imager Radiometer Suite (VIIRS), may be able to ameliorate this issue. Future studies aiming to map dust-emission sources using an approach similar to the Walker et al. (2009) methodology should consider use of these higher-resolution satellite platforms to enhance dust-emission potential associations to various landscape attributes. Additionally, MODIS is currently operating past its initial design life of six years, so continued dependency on image acquisition from MODIS could pose a risk. The significant drawback to any new satellite capability, however, is a substantial reduction in the length of the record. VIIRS, for example, was launched in 2011 while MODIS was launched in 2000 and can therefore provide a longer record of continuous data.

A second limitation of the approach used in this study is the fact that MODIS visible-channel imagery can only be acquired during daylight hours. Satellite orbit timing could also affect dust-level estimations as diurnal patterns of blowing dust exist due to the heterogeneous heating of the atmosphere and land surface (Stout 2010). If a satellite does not pass over the area of interest during times with the greatest frequency of dust storms, it could miss the dust events (particularly if short lived) entirely. For example, Orgill and Sehmel (1976) examined dust storms reported in weather station data in the United States and determined that the majority of dust storms in the region occur between 1200 and 2000. This time period only partially aligns with MODIS satellite pass over in the Southwest United States, which is always between 1700 and 2100 UTC.

The team of CRREL analysts will conduct future research efforts using the Walker et al. (2009) methodology to better understand how temporal and spatial patterns of dust emissivity relate to local climate patterns and regional geomorphology. The analysts who participated in this study will build a dust-plume-head point source database using the Walker et al. (2009) technique and 16 continuous years of MODIS data for U.S. and Mexico desert regions. Findings from this future study will help guide data interpretation and collaborative mapping approaches.

References

Adams, D. K., and A. C. Comrie. 1997. The North American Monsoon. *Bulletin of the American Meteorological Society* 78 (10): 2197–2213.

Al-Hemoud, A., M. Al-Sudairawi, S. Neelamanai, A. Naseeb, and W. Behbehani. 2017. Socioeconomic Effect of Dust Storms in Kuwait. *Arabian Journal of Geosciences* 10 (1): 1–9.

Affleck, R., P. Seman, M. Deegan, R. Freeman, and S. Sargand. 2011. *Documenting Lessons Learned in Afghanistan Concerning Design and Construction Challenges.* ERDC/CRREL TR-11-02. Hanover, NH: U.S. Army Engineer Research and Development Center.

Bacon, S. N., and E. V. McDonald. 2016. Regional Distribution of Salt-Rich Dust Across Southwest Asia Based on Predictive Soil-Geomorphic Mapping Techniques. In *Military Geosciences and Desert Warfare*, ed. E. V. McDonald and T. Bullard, 237–56. New York, NY: Springer.

Baddock, M. C., J. E. Bullard, and R. G. Bryant. 2009. Dust Source Identification Using MODIS: A Comparison of Techniques Applied to the Lake Eyre Basin, Australia. *Remote Sensing of Environment* 113 (7): 1511–28.

Baddock, M. C., P. Ginoux, J. E. Bullard, and T. E. Gill. 2016. Do MODIS-Defined Dust Sources Have a Geomorphological Signature? *Geophysical Research Letters* 43 (6): 2606–2613.

Breckle, S.-W. 2007. Flora and Vegetation of Afghanistan. *Basic and Applied Dryland Research* 1 (2): 155–94.

Brindley, H., P. Knippertz, C. Ryder, and I. Ashpole. 2012. A Critical Evaluation of the Ability of the Spinning Enhanced Visible and Infrared Imager (SEVIRI) Thermal Infrared Red-Green-Blue Rendering to Identify Dust Events: Theoretical Analysis. *Journal of Geophysical Research: Atmospheres* 117 (D7): D07201. doi:10.1029/2011JD017326.

Bullard, J. E., M. C. Baddock, G. McTainsh, and J. Leys. 2008. Sub-Basin Scale Dust Source Geomorphology Detected Using MODIS. *Geophysical Research Letters* 35 (15): L15404. doi:10.1029/2008GL033928.

Bullard, J. E., S. P. Harrison, M. C. Baddock, N. Drake, T. E. Gill, G. McTainsh, and Y. Sun. 2011. Preferential Dust Sources: A Geomorphological Classification Designed for Use in Global Dust-Cycle Models. *Journal of Geophysical Research* 116 (F4): F04034. doi:10.1029/2011JF002061.

De Longueville, F., Y.-C. Hountondji, S. Henry, and P. Ozer. 2010. What Do We Know About Effects of Desert Dust on Air Quality and Human Health in West Africa Compared to Other Regions? *Science of the Total Environment* 409 (1): 1–8.

EPA (U.S. Environmental Protection Agency). 2012. Ecoregions of North America. http://maps4.arcgisonline.com/ArcGIS/rest/services/DOI/EPA_Ecoregions_of_North_America/MapServer (accessed 23 September 2016).

Gillette, D. A. 1999. A Qualitative Geophysical Explanation for Hot Spot Dust Emitting Source Regions. *Contributions to Atmospheric Physics* 72 (1): 67–77.

Ginoux, P., J. M. Prospero, T. E. Gill, N. C. Hsu, and M. Zhao. 2012. Global-Scale Attribution of Anthropogenic and Natural Dust Sources and Their Emission Rates Based on MODIS Deep Blue Aerosol Products. *Reviews of Geophysics* 50 (3): RG3005. doi:10.1029/2012RG000388.

Goudie, A., and N. J. Middleton. 2006. *Desert Dust in the Global System*. New York: Springer.

Huang, J., T. Wang, W. Wang, Z. Li, and H. Yan. 2014. Climate Effects of Dust Aerosols over East Asian Arid and Semiarid Regions. *Journal of Geophysical Research: Atmospheres* 119 (19): 11,398–11,416.

Jafari, R., and M. Malekian. 2015. Comparison and Evaluation of Dust Detection Algorithms Using MODIS Aqua/Terra Level 1B Data and MODIS/OMI Dust Products in the Middle East. *International Journal of Remote Sensing* 36 (2): 597–617.

Khaurin, H. H. 2003. *Trees and Bushes of Afghanistan.* Kabul, Afghanistan: Food and Agriculture Organization, Forest National Coordination Office. http://eafghanag.ucdavis.edu/c_livestock/Man_For_Fodder_Trees_Afghan_FAO.pdf (accessed 01 February 2017).

King, J., V. Etyemezian, M. Sweeney, B. J. Buck, and G. Nikolich. 2011. Dust Emission Variability at the Salton Sea, California, USA. *Aeolian Research* 3 (1): 67–79.

Knippertz, P., and J.-B. W. Stuut, ed. 2014. *Mineral Dust: A Key Player in the Earth System*. New York: Springer.

Lary, D. J., A. H. Alavi, A. H. Gandomi, and A. L. Walker. 2016. Machine Learning in Geosciences and Remote Sensing. *Geoscience Frontiers* 7 (1): 3–10.

Legrand, M., C. N'doume, and I. Jankowiak. 1994. Satellite-Derived Climatology of the Saharan Aerosol. In *Passive Infrared Remote Sensing of Clouds and the Atmosphere II*, ed. D. K. Lynch, 127–35. Proceedings of SPIE International Society of Optical Engineering.

Lensky, I. M., and D. Rosenfeld. 2008. Clouds-Aerosols-Precipitation Satellite Analysis Tool (CAPSAT). *Atmospheric Chemistry and Physics* 8 (22): 6739–53.

Liu, M., D. L. Westphal, A. L. Walker, T. R. Holt, K. A. Richardson, and S. D. Miller. 2007. COAMPS real-Time Dust Storm Forecasting During Operation Iraqi Freedom. *Weather and Forecasting* 22 (1): 192–206.

Mahowald, N. M., A. R. Baker, G. Bergametti, N. Brooks, R. A. Duce, T. D. Jickells, N. Kubilay, J. M. Prospero, and I. Tegen. 2005. Atmospheric Global Dust Cycle and Iron Inputs to the Ocean. *Global Biogeochemical Cycles* 19 (4): GB4025. doi:10.1029/2004GB002402.

Mahowald, N., S. Albani, J. F. Kok, S. Engelstaeder, R. Scanza, D. S. Ward, and M. G. Flanner. 2014. The Size Distribution of Desert Dust Aerosols and Its Impact on the Earth System. *Aeolian Research* 15:53–71.

McDonald, E., S. Bacon, S. Baker, R. Amit, J. L. Antinao, M. Berli, T. Bullard, T. Caldwell, O. Crouvi, and Y. Enzel. 2013. *Integrated Desert Terrain Forecasting for Military Operations.* Reno, NV: Desert Research Institute.

Middleton, N. J. 2017. Desert Dust Hazards: A Global Review. *Aeolian Research* 24:53–63.

Miller, S. D. 2003. A Consolidated Technique for Enhancing Desert Dust Storms with MODIS. *Geophysical Research Letters* 30 (20): 2071. doi:10.1029/2003GL018279.

Moridnejad, A., N. Karimi, and P. A. Ariya. 2015. A New Inventory for Middle East Dust Source Points. *Environmental Monitoring and Assessment* 187 (9): 1–11.

Muhs, D. R., J. M. Prospero, M. C. Baddock, and T. E. Gill. 2014. Identifying Sources of Aeolian Mineral Dust: Present and Past. In *Mineral Dust*, ed. P. Knippertz and J.-B. W. Stuut, 51–74. New York: Springer.

Nadolski, V. L. 1998. *Automated Surface Observing System (ASOS) User's Guide*. Silver Spring, MD: National Oceanic and Atmospheric Administration. http://www.nws.noaa.gov/asos/pdfs/aum-toc.pdf (accessed 11 November 2016).

Okin, G. S., J. E. Bullard, R. L. Reynolds, J.-A. C. Ballantine, K. Schepanski, M. C. Todd, J. Belnap, M. C. Baddock, T. E. Gill, and M. E. Miller. 2011. Dust: Small-Scale Processes with Global Consequences. *EOS, Transactions American Geophysical Union* 92 (29): 241.

Orgill, M. M., and G. A. Sehmel. 1976. Frequency and Diurnal Variation of Dust Storms in the Contiguous USA. *Atmospheric Environment* 10 (10): 813–25.

Prospero, J. M., P. Ginoux, O. Torres, S. E. Nicholson, and T. E. Gill. 2002. Environmental Characterization of Global Sources of Atmospheric Soil Dust Identified with the NIMBUS 7 Total Ozone Mapping Spectrometer (TOMS) Absorbing Aerosol Product. *Reviews of Geophysics* 40 (1): 2-1–2-31.

Rashki, A., D. G. Kaskaoutis, A. S. Goudie, and R. A. Kahn. 2013. Dryness of Ephemeral Lakes and Consequences for Dust Activity: The Case of the Hamoun Drainage Basin, Southeastern Iran. *Science of the Total Environment* 463 (1): 552–564.

Ravi, S., P. D'Odorico, D. D. Breshears, J. P. Field, A. S. Goudie, T. E. Huxman, J. Li, G. S. Okin, R. J. Swap, A. D. Thomas, S. Van Pelt, J. J. Whicker, and T. M. Zobeck. 2011. Aeolian Processes and the Biosphere. *Reviews of Geophysics* 49 (3): RG3001. doi:10.1029/2010RG000328.

Rushing, J. F., and J. S. Tingle. 2006. *Dust Control Field Handbook: Standard Practices for Mitigating Dust on Helipads, Lines of Communication, Airfields, and Base Camps.* ERDC/GSL SR-06-7. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Rushing, J. F., A. Harrison, and J. S. Tingle. 2005. *Evaluation of Application Methods and Products for Mitigating Dust for Lines-of-Communication and Base Camp Operations.* ERDC/GSL TR-05-9. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Shepherd, G., E. Terradellas, A. Baklanov, U. Kang, W. Sprigg, S. Nickovic, A. D. Boloorani, A. Al-Dousari, S. Basart, and A. Benedetti. 2016. *Global Assessment of Sand and Dust Storms.* Nairobi: United Nations Environment Programme.

Shinn, E. A., G. W. Smith, J. M. Prospero, P. Betzer, M. L. Hayes, V. Garrison, and R. T. Barber. 2000. African Dust and the Demise of Caribbean Coral Reefs. *Geophysical Research Letters* 27 (19): 3029–32.

Skiles, S. M., T. H. Painter, J. Belnap, L. Holland, R. L. Reynolds, H. L. Goldstein, and J. Lin. 2015. Regional Variability in Dust-on-Snow Processes and Impacts in the Upper Colorado River Basin. *Hydrological Processes* 29 (26): 5397–5413.

Sprigg, W. A., S. Nickovic, J. N. Galgiani, G. Pejanovic, S. Petkovic, M. Vujadinovic, A. Vukovic, M. Dacic, S. DiBiase, A. Prasad, and H. El-Askary. 2014. Regional Dust Storm Modeling for Health Services: The Case of Valley Fever. *Aeolian Research* 14:53–73.

Stout, J. E. 2010. Diurnal Patterns of Blowing Sand. *Earth Surface Processes and Landforms* 35 (3): 314–18.

Sweeney, M. R., and Joseph A. Mason. 2013. Mechanisms of Dust Emission from Pleistocene Loess Deposits, Nebraska, USA. *Journal of Geophysical Research: Earth Surface* 118 (3): 1460–71.

Sweeney, M. R., E. V. McDonald, and V. Etyemezian. 2011. Quantifying Dust Emissions from Desert Landforms, Eastern Mojave Desert, USA. *Geomorphology* 135 (1–2): 21–34.

Sweeney, M. R., H. Lu, M. Cui, J. A. Mason, H. Feng, and Z. Xu. 2016. Sand Dunes as Potential Sources of Dust in Northern China. *Science China Earth Sciences* 59 (4): 760–69.

Tong, D. Q., M. Dan, T. Wang, and P. Lee. 2012. Long-Term Dust Climatology in the Western United States Reconstructed from Routine Aerosol Ground Monitoring. *Atmospheric Chemistry and Physics* 12 (11): 5189–5205.

Torres, O., P. K. Bhartia, J. R. Herman, Z. Ahmad, and J. Gleason. 1998. Derivation of Aerosol Properties from Satellite Measurements of Backscattered Ultraviolet Radiation: Theoretical Basis. *Journal of Geophysical Research: Atmospheres* 103 (D14): 17099–110.

Walker, A. L., M. Liu, S. D. Miller, K. A. Richardson, and D. L. Westphal. 2009. Development of a Dust Source Database for Mesoscale Forecasting in Southwest Asia. *Journal of Geophysical Research* 114 (D18): D18207. doi:10.1029/2008JD011541.

Wang, F., X. Zhao, C. Gerlein-Safdi, Y. Mu, D. Wang, and Q. Lu. 2017. Global Sources, Emissions, Transport and Deposition of Dust and Sand and Their Effects on the Climate and Environment: A Review. *Frontiers of Environmental Science and Engineering* 11 (1): 1–9. doi:10.1007/s11783-017-0904-z.

Webb, N. P., A. Chappell, C. L. Strong, S. K. Marx, and G. H. McTainsh. 2012. The Significance of Carbon-Enriched Dust for Global Carbon Accounting. *Global Change Biology* 18 (11): 3275–78.

Zhang, X., D. Q. Tong, G. Wu, X. Wang, A. Xiu, Y. Han, T. Xu, S. Zhang, and H. Zhao. Forthcoming. Identification of Dust Sources and Hotspots in East Asia During 2000–2015: Implications for Numerical Modeling and Forecasting. *Atmospheric Chemistry and Physics Discussions*. doi:10.5194/acp-2016-681.

Appendix A: Obtaining ASOS Data

To obtain weather station data for a particular area of interest, go to https://mesonet.agron.iastate.edu/request/download.phtml?network=AZ_ASOS. Look for "Select Network" at the top of the page, choose a region of interest, and click "Switch to Network." Section 1, "Select Station/Network by Clicking on Location," shows a map of the selected area with red markers denoting the location of each ASOS station in the region. Above the map, each station is listed alphabetically under "Sort Available Stations." From this list, choose several stations that cover the area of interest and click "Add Selected." It is not necessary to download data from every available station, particularly in regions where an abundance exist; but stations both within and immediately outside of the area of concern should be downloaded to obtain comprehensive coverage of the region. The chosen stations will appear in the "Selected Stations" box. In section 2, "Select from Available Data," select "All Available" to include all measurements and observations from the ASOS stations. In section 3, "Specific Date Range," select the desired range of interest. In section 4, "Time Zone of Observation Times," select UTC to avoid potential problems with regions that span across multiple time zones. In section 5, "Download Options," select tab delimited, include the latitude and longitude data, and change "View result data in web browser" to "Save result data to file on computer." After making these selections, click "Get Data."

Once downloaded, the data will need to undergo some minor processing to alter the date and time stamps associated with each entry and to remove unwanted fields. Software engineers at Atmospheric and Environmental Research, Inc., developed a Python script to handle these processing steps. Use the Python script entitled "ASOS_Processing.py" (included below), and type the following command: `python ASOS_Processing.py /path/to/directory`, where `/path/to/directory` is replaced with the correct path name. Each processed weather station dataset produces a text (.txt) and a comma-separated values (.csv) file.

ASOS Processing Python Script:

```
'''
---
Title: ASOS_Processing.py
Date Created: December 2015
Updated: March 2017
Outputs: For each weather station, a text file with all hourly reports processed -->
<inputfile>_PROCESSED.txt

For all weather stations processed in a single batch, a summary .xls file with only reports
from "events" (defined below) included --> ASOS_STATS_events_only.xls
---


#-------------------------------------------------------------------------------
# PURPOSE
#-------------------------------------------------------------------------------

ASOS_Processing.py processes weather station data files downloaded from the Iowa
Environmental Mesonet ASOS Network. The URL for the IEM ASOS archive is:
https://mesonet.agron.iastate.edu/request/download.phtml?network=AZ_ASOS

This script can process a single file or a batch of data files contained in the same
directory. This script will produce 1) a text file of processed data for each raw data file
it is given and 2) a summary .xls file that compiles all reports associated with decreased
visibility or obscurations reported.


#-------------------------------------------------------------------------------
# SUMMARY
#-------------------------------------------------------------------------------

ASOS_processing.py reads data files downloaded from the IEM ASOS Network and processes them
by scanning for missing values, changing the data type of the time stamp field, removing
unwanted fields, and searching for the occurrence of visibility obscurations and
thunderstorms (which are logged in a new field).  The processed data is saved in a new tab
delimited text file with the input file name appended with the suffix _PROCESSED.txt.

For each weather station data file:
input file = my_data.txt
output file = my_data_PROCESSED.txt



Events summary file compilation:

Events are defined as reports that contain decreased visibility conditions (< 7 nautical
miles), reports of obscuration (blowing dust (BLDU), dust storm (DS), sand storm (SS), dust
whorls or dust devils (PO), haze (HZ), fog/mist, or smoke), or a combination of the two.


event_code          Obscuration and Visibility Condition

0.0     =           no obscurations and high visibility
1.0     =           low visibility, no obscuration reported
2.0     =           low visibility and BLDU, DS, SS, PO
2.1     =           high visibility and BLDU, DS, SS, PO
3.0     =           low visibility and haze
3.1     =           high visibility and haze
8.0     =           low visibility and fog/mist
8.1     =           high visibility and fog/mist
9.0     =           low visibility and smoke
9.1     =           high visibility and smoke

All reports are assigned an event_code. All reports with an event code that is not equal to
0.0 are compiled to remove duplicate reports of the same event on the same day. Therefore,
```

for a given station, only one report of an event is recorded per day (this is for
calculating statistics of the occurrence of events).

This compilation is saved in: ASOS_STATS_events_only.xls

```
#-------------------------------------------------------------------------------
# USAGE
#-------------------------------------------------------------------------------
```

To run this processing script, place a copy of this code in a directory with the data
file(s) that need to be processed. In Terminal, navigate to the directory that contains the
script and data file(s), then enter this command:

python ASOS_Processing.py /path/to/directory/with/data/files

Where you will replace the /path/to/directory... with the correct path name. The script
will print updates and information to the terminal window about the processing. This script
will create, use, and then delete a temporary file in the directory that holds the script
and your data file to be processed. The processed output files will be saved to this same
working directory.

```python
'''
import pandas as pd
import os
import sys
import datetime

print sys.argv[-1]
directory_in = sys.argv[-1]
# read directory path from launch command
# check to make sure a directory was provided
if directory_in == '*.py':
print''
print 'You forgot to include the directory path in the command'
print 'Please run again and tell me where to look for the zip files.'
print''
sys.exit()

files = os.listdir(directory_in)
#store the names of all files in directory in a list "files"
print files

nfiles = len(files)
print nfiles

aa = 0
processed = 0

while aa < nfiles:
    current_file = files[aa]
   # grab a file in the list
    print "The current file is: " +current_file
    suffix = current_file[-4:]
   # cut the last 4 characters off the file name to check the type of file
    if suffix != '.txt':
        print ''
         print 'NOT A DATA FILE'
        # if it is not a .txt, do nothing
         print 'Move to the next one...'
         print ''
         aa = aa+1

    else :
number = aa+1
print 'Data File --> Statring  To Process ' +str(number) +' of ' +str(nfiles)
# if it is a .zip, unzip it to the current directory
```

```
raw_input_file = directory_in+'/'+current_file
output_file = 'ASOS_STATS_events_only.xls'
# generate output file name

# Need to strip the first 5 lines of the input file downloaded from IEM ASOS archive
numline=5
# 5 lines to skip
p=""
o=open("temp_output.txt","a")
f=open(raw_input_file)
for i in range(numline):
f.next()
for line in f:
if p:
o.write(p)
p=line
f.close()
o.close()

# open data file with first 5 lines removed to read the data, skip spaces after tab
delimiters
df_in = pd.read_csv('temp_output.txt', delimiter='\t', skipinitialspace=True,)
# read data from file into pandas Data Frame
os.remove('temp_output.txt')
# delete the intermediate file with the top 5 lines stripped off
print 'Beginning Data Processing'

# ******************************************************************
# ******************************************************************

# DATA PROCESSING - carried out in order of the parameter columns from input data file
# Station Name --> Add "K" to station text string to match shapefile names.
add_str = 'K'
for i in range(0,len(df_in.index)):
existing_str = df_in.loc[i,'station']
df_in.set_value(i, 'station', add_str+existing_str)
# Time Stamp Processing --> Convert 'valid' from string to type = datetime

df_in['valid'] = pd.to_datetime(df_in['valid'])




# For Numeric Data -- Scan for missing values (marked with "M" by IEM ASOS data
interpreter) and assign a value of "null" (without "") if found
# Also add hooks for unit conversion

# ***********************************
# ***********************************

# Define a function to look at columns that should have numeric values, but they are mixed
in with strings ("M") from the IEM ASOS interpreter for missing values
def is_number(s):
try:
float(s)
return True
except ValueError:
return False

# ***********************************
# ***********************************

# Check the Data, Replace M's from IEM ASOS interp. with '' (no data) so that ArcGIS reads
the column in as a float and the empty cells will become <Null> values
#
df_in['hour'] = pd.Series(0, index=df_in.index)
 # create the new column
df_in['day'] = pd.Series(0, index=df_in.index)
```

```
# create the new column
df_in['month'] = pd.Series(0, index=df_in.index)
# create the new column
df_in['year'] = pd.Series(0, index=df_in.index)
# create the new column
df_in['obscure'] = pd.Series('none', index=df_in.index)
# create the new column
df_in['Tstorm'] = pd.Series('none', index=df_in.index)
# create the new column
df_in['dailypre'] = pd.Series(0.0, index=df_in.index)
# create the new column
df_in['event'] = pd.Series('none', index=df_in.index)
# create the new column
df_in['event_code'] = pd.Series(0.0, index=df_in.index)
# create the new column
df_in['vis_check'] = pd.Series(0.0, index=df_in.index)
# create the new column
df_in['dup_report'] = pd.Series(0, index=df_in.index)
# create the new column

# Temperature data processing
for i in range(0,len(df_in.index)):
temp = df_in.loc[i,'tmpf']
check = is_number(temp)
if check != True:
df_in.set_value(i,'tmpf', '')

# Dew Point data processing
for i in range(0,len(df_in.index)):
dewpt = df_in.loc[i,'dwpf']
check = is_number(dewpt)
if check != True:
df_in.set_value(i,'dwpf', '')

# Relative Humidity data processing
for i in range(0,len(df_in.index)):
RH = df_in.loc[i,'relh']
check = is_number(RH)
if check != True:
df_in.set_value(i,'relh', '')

# Wind Direction data processing
for i in range(0,len(df_in.index)):
win_dir = df_in.loc[i,'drct']
check = is_number(win_dir)
if check != True:
df_in.set_value(i,'drct', '')

# Wind Speed data processing
for i in range(0,len(df_in.index)):
win_spd = df_in.loc[i,'sknt']
check = is_number(win_spd)
if check != True:
df_in.set_value(i,'sknt', '')

# 1 Hour Precipitation data processing
for i in range(0,len(df_in.index)):
precip1hr = df_in.loc[i,'p01i']
check = is_number(precip1hr)
if check != True:
df_in.set_value(i,'p01i', '')

print '...  Making Progress'



# Pressure Altimeter data processing
```

```
for i in range(0,len(df_in.index)):
test_val = df_in.loc[i,'alti']
check = is_number(test_val)
if check != True:
df_in.set_value(i,'alti', '')

# Sea Level Pressure data processing
for i in range(0,len(df_in.index)):
test_val = df_in.loc[i,'mslp']
check = is_number(test_val)
if check != True:
df_in.set_value(i,'mslp', '')

# Visibility data processing
for i in range(0,len(df_in.index)):
vis = df_in.loc[i,'vsby']
check = is_number(vis)
if check != True:
df_in.set_value(i,'vis_check', 99.0)
else:
df_in.set_value(i,'vis_check', vis)

# Gust Wind Speed data processing
for i in range(0,len(df_in.index)):
gust_val = df_in.loc[i,'gust']
#print gust_val
check = is_number(gust_val)
if check != True:
df_in.set_value(i,'gust', '')

print '... ... Almost Done'
print ''

# search METAR string for different obscuration codes and store them in a new column called
"obscure"

for i in range(0,len(df_in.index)):
input = df_in.loc[i,'valid']
year = input.year
month = input.month
day = input.day
hour = input.hour
df_in.set_value(i,'year', year)
df_in.set_value(i,'month', month)
df_in.set_value(i,'day', day)
df_in.set_value(i,'hour', hour)


# Loop through the METAR strings, ask if any of the obscuration or storm codes are in each
# Individual METAR string, if so, change the value in the "obscure" or "Tstorm" fields
for i in range(0,len(df_in.index)):
metar = df_in.loc[i,'metar']
# Get row i of METAR
BLDU = metar.find('BLDU')
# Search for "BLDU" and log the location
DSNT =  metar.find('DSNT')
DS = metar.find('DS')
ALQDS = metar.find('ALQDS')
RMK = metar.find('RMK')
PO = metar.find('PO')
HZ = metar.find('HZ')
fog = metar.find('FG')
smoke = metar.find('FU')
SA = metar.find('SA')
mist = metar.find('BR')
tstorm = metar.find('TS')
zap = metar.find('LTG')
```

```
fropa = metar.find('FROPA')
dp = metar.find(' 70')

if BLDU != -1:
df_in.set_value(i,'obscure', 'Blowing Dust')
df_in.set_value(i,'event', 'dust/sand obs')
# Continue adding the text strings
elif PO != -1:
df_in.set_value(i,'obscure', 'Dust Devils')
df_in.set_value(i,'event', 'dust/sand obs')
elif HZ != -1:
df_in.set_value(i,'obscure', 'Haze')
df_in.set_value(i,'event', 'haze')
elif fog != -1:
df_in.set_value(i,'obscure', 'Fog')
df_in.set_value(i,'event', 'fog/mist')
elif smoke != -1:
df_in.set_value(i,'obscure', 'Smoke')
df_in.set_value(i,'event', 'smoke')
elif SA != -1:
df_in.set_value(i,'obscure', 'Sandstorm')
df_in.set_value(i,'event', 'dust/sand obs')

elif mist != -1:
df_in.set_value(i,'obscure', 'Mist')
df_in.set_value(i,'event', 'fog/mist')
# The dust storms code (DS) looks the same as other codes when searched for this way. Check
#the location of DS and the things around it to see if it is actually a
#dust storm or just a part of another code statement.
elif DS != -1:
if DS < RMK:
# DS is a part of DSNT --> ignore!
df_in.set_value(i,'obscure', 'Duststorm')
df_in.set_value(i,'event', 'dust/sand obs')
# Now look for info on thunderstorms and lightning
if tstorm != -1:
df_in.set_value(i,'Tstorm', 'Thunderstorm 0-10 miles')
elif zap != -1 and tstorm == -1:
df_in.set_value(i,'Tstorm', 'Thunderstorm 10-30 miles')
# Look for a remark about a passing front
if fropa != -1:
if tstorm == -1 and zap == -1:
df_in.set_value(i,'Tstorm', 'Frontal Passage no Tstorm')
else:
df_in.set_value(i,'Tstorm', 'Frontal Passage with Tstorm')
# get the 24 hour precipitation total
if dp != -1:
if dp > RMK:
# Check to make sure it is in the remarks section
a = metar[dp+2:dp+6]
# Grab the daily precipitation data in 1/100th of an inch
if is_number(a) == True:
# Make sure you have 4 numbers in the string
b = float(a)
c = b/100
df_in.set_value(i,'dailypre', c)

# Define new column "event_code" to make it easy to sort the data based on observations of
vision, dust, fog, etc.
# Code for different combinations of low visibility (< 7 nautical miles) and obscurations

# 0 = no obscurations and high visibility
# 1.0 = low visibility, no obscuration reported
# 2.0 = low visibility and BLDU, DS, SS, PO
# 2.1 = high visibility and BLDU, DS, SS, PO
# 3.0 = low visibility and haze
# 3.1 = high visibility and haze
```

```python
# 8.0 = low visibility and fog/mist
# 8.1 = high visibility and fog/mist
# 9.0 = low visibility and smoke
# 9.1 = high visibility and smoke


for i in range(0,len(df_in.index)):
event_str = df_in.loc[i,'event']
vis = df_in.loc[i,'vis_check']
if vis > 7.0:
if event_str == 'none':
df_in.set_value(i,'event_code', 0.0)
elif event_str == 'dust/sand obs':
df_in.set_value(i,'event_code', 2.1)
elif event_str == 'haze':
df_in.set_value(i,'event_code', 3.1)
elif event_str == 'fog/mist':
df_in.set_value(i,'event_code', 8.1)
elif event_str == 'smoke':
df_in.set_value(i,'event_code', 9.1)
elif vis <= 7.0:
if event_str == 'none':
df_in.set_value(i,'event_code', 1.0)
elif event_str == 'dust/sand obs':
df_in.set_value(i,'event_code', 2.0)
elif event_str == 'haze':
df_in.set_value(i,'event_code', 3.0)
elif event_str == 'fog/mist':
df_in.set_value(i,'event_code', 8.0)
elif event_str == 'smoke':
df_in.set_value(i,'event_code', 9.0)

print ''
print 'Data Processed --> Saving File'
print ''

# Clean Up Processing – A search for mistaken obscurations

for i in range(0,len(df_in.index)):
#index = df_in.index[i]
# Create an index
metar = df_in.loc[i,'metar']
# Get row i of METAR
PO = metar.find('PO')
# Look for Dust Devil reports
POINT = metar.find('POINT')
# Look for PO occurring as part of DEWPOINT
PORAI = metar.find('PORAI')
# Look for PO occurring as part of TEMPORARIRILY

if PO != -1:
# Scan for PO being part of a different report, if so, set event_code to zero
if PO == POINT:
df_in.set_value(i,'event_code', 0.0)
df_in.set_value(i,'event', 'station down')
print metar
elif PO == PORAI:
df_in.set_value(i,'event_code', 0.0)
df_in.set_value(i,'event', 'station down')
print metar

# OUTPUT Individual Station File --------------------
df_station = df_in
df_station = df_station.drop(df_station.columns[[13, 14, 15, 16, 17, 18, 19, 20]], axis=1)
# Remove the unwanted/unnecessary columns from the data
output_station_file = current_file[:-4]+'_PROCESSED.txt'
# generate output file name
```

```
df_station.to_csv(output_station_file, sep='\t', index=False)
# Save station data to individual _PROCESSED.xls file
# ----------------------------------------------------

# Prepare compiled events only output

# Trim the output to the reports with Events only (i.e. event_code != to 0)
df_in = df_in[df_in.event_code != 0]

# Remove multiple reports of same day events

for i in range(0,len(df_in.index)-1):
index1 = df_in.index[i]
index2 = df_in.index[i+1]
day1 = df_in.loc[index1,'day']
day2 = df_in.loc[index2,'day']
month1 = df_in.loc[index1,'month']
month2 = df_in.loc[index2,'month']
event1 = df_in.loc[index1,'event_code']
event2 = df_in.loc[index2,'event_code']

if day1 == day2:
if month1 == month2:
if event1 == event2:
df_in.set_value(index1,'dup_report', 1)

df_out = df_in
df_out = df_out.drop(df_out.columns[[13, 14, 15, 16, 17, 18, 19, 20]], axis=1)
# Remove the unwanted/unnecessary columns from the data
df_out = df_out[df_out.dup_report != 1]

event_counter = len(df_out.index)

print '%d Events Observed' %event_counter
print ''
df_out.to_csv(output_file, sep='\t', index=False, mode='a', header=False)
# save the data frame to a tab delimited text file
print ''
print 'File Saved.  !'
print ''
aa = aa + 1 # increment the counter to grab the next file

print '-------------- FINISHED ----------------'
```

Appendix B: Creating MODIS Imagery

To download MODIS imagery, create a folder that contains the Python script entitled "MODIS_Dust_Detection.py" (included below). All downloaded images will automatically save to this folder. Type the following command: `python MODIS_Dust_Detection.py`. A prompt will appear asking for the southern latitude, northern latitude, western longitude, and eastern longitude boundaries of the area of interest. Note, a negative latitude refers to the Southern Hemisphere while a negative longitude refers to the Western Hemisphere. A second prompt will ask for a start and end date for which the satellite imagery will be downloaded. Using the list of dates generated through inspecting the ASOS weather station data, enter a date (in yyyy/mm/dd format) associated with a single dust event. Downloading MODIS images can be time consuming, so users should not download images for a period greater than three days at one time. Instead, break up longer multiday dust events into two separate downloads. Finally, a prompt to create an output file name will be displayed. A single day may contain several MODIS images, so files should be named according to the dates they are associated with. An example of user input is shown below.

Python Script Prompts	Example User Inputs
Enter Southern Latitude Bound	30
Enter Northern Latitude Bound	40
Enter Western Latitude Bound	-120
Enter Eastern Latitude Bound	-105
Enter Beginning Date (e.g. 2003/12/31)	2016/02/01
Enter Ending Date (e.g. 2003/12/31)	2016/02/03
Enter Name for Output File	February_1-3_2016

There are two known processing issues that exist when downloading MODIS imagery. The first involves an error message that states that the download has "Timed Out," which is likely a result of downloading images for a date range that is too long. When this error occurs, split any timed out multiday events into two separate downloads. Alternatively, if a single-day event times out, try downloading it a second time. A second error message, "Too Many Users," is a result of too many individuals downloading MODIS data from NASA (National Aeronautics and Space Administration) at one time. This error message is rare; but when encountered, users should try to redownload the images later.

MODIS Dust Detection Python Script

```
# -*- coding: utf-8 -*-
# modis_dust_detection.py is a script built to download, process, and
# resample MODIS imagery for a specified area of interest and period of time.
# The immediate purpose being for the detection of dust in the atmosphere.
# The final result of the script are a true color composite and dust specific
# two false color composites, using the EU-METSAT and Navy Research Labora-
# tory (Miller 2003) algorithms. The output is stored in GeoTIFF format.
#
# Instructions:
#-Execute script from target directory for output files
#-User will be prompted for desired Latitude/Longitude bounds, start/end
# dates (year/month/day), and a name for the scene.
#
# Jeff Picton
# jpicton@aer.com
# Atmospheric and Environmental Research Inc.
# 2015


from ftplib import FTP
from shapely.geometry import Polygon
import datetime
from pyhdf import SD
from scipy.interpolate import griddata
import numpy as np
import gdal
import osr
import sys
import os


np.seterr(invalid='ignore')



# define granule class
class granule:
  def __init__(self,file03,file02,starttime,arc,orbit,daynight,lon,lat):
      self.File03    = file03
      self.File02    = file02
      self.StartTime = starttime
      self.Archive   = arc
      self.Orbit     = orbit
      self.DayNight  = daynight
      self.Lon       = lon
      self.Lat       = lat

# get user inputs
def getuserinput():
    # get input parameters
    # Hard wire a working window that produces a google earth-able
    # size image by commenting out[#] the raw_input() statements
    lat1 = raw_input('Enter Southern Latitude bound:  ')
    #lat1 = '30'
```

```
    lat2 = raw_input('Enter Northern Latitude bound:  ')
    #lat2 = '40'
    lon1 = raw_input('Enter Western Longitude bound:  ')
    #lon1 = '-120'
    lon2 = raw_input('Enter Eastern Longitude bound:  ')
    #lon2 = '-105'
    date1 = raw_input('Enter beginning date (e.g. 2003/12/31):  ')
    date2 = raw_input('Enter ending date (e.g. 2003/12/31):  ')
    aoi = raw_input('Enter name for output files:  ')


    # save imagery info to log file
    ff = open('imagery_log.txt', 'ab')
    # open existing file to append or create a new file
    ff.write(lat1+'\t')   # write the data and end with a tab
    ff.write(lat2+'\t')
    ff.write(lon1+'\t')
    ff.write(lon2+'\t')
    ff.write(date1+'\t')
    ff.write(date2+'\t')
    ff.write(aoi+'\n')
    ff.close()   # close the file

    # convert to correct data types
    try:
        latbnds = [float(lat1), float(lat2)]
        lonbnds = [float(lon1), float(lon2)]
        date1 = datetime.datetime.strptime(date1,'%Y/%m/%d')
        date2 = datetime.datetime.strptime(date2,'%Y/%m/%d')
    except:
        print('Invalid input found - Please try again')
        sys.exit()
    if latbnds[0]>latbnds[1]:
        print('Upper Latitude bound must be greater than lower ')
        sys.exit()
    if latbnds[0]>latbnds[1]:
        print('Upper Longitude bound must be greater than lower')
        sys.exit()
    if date1>date2:
        print('End date cannot be before start date')
    return latbnds,lonbnds,date1,date2,aoi

# return list of granules that cover aoi
def findgranules(latbnds,lonbnds,date1,date2):
    # enumerate dates
    days = [date1 + datetime.timedelta(n) for n in range((date2-
date1).days+1)]
    # define platform variables

    # -----  Modified because TERRA satellite in safe mode starting 2/19/2016

    TERRA_down_date = '2016/02/19'
    TERRA_down_date = datetime.datetime.strptime(TERRA_down_date,'%Y/%m/%d')
    AQUA_start_date = '2002/08/13'
    AQUA_start_date = datetime.datetime.strptime(AQUA_start_date,'%Y/%m/%d')
    test_TERRA = (date1-TERRA_down_date).days
    test_AQUA = (date1-AQUA_start_date).days
```

```
    if test_TERRA > 0:
        platforms = ['AQUA']
        prefixes = {'AQUA':'MYD'}
        print ''
        print 'Only Use AQUA because TERRA is down.'
        print ''
    elif test_AQUA < 0:
        platforms = ['TERRA']
        prefixes = {'TERRA':'MOD'}
        print ''
        print 'Only Use TERRA because AQUA was not operational yet.'
        print ''
    else:
        platforms = ['TERRA','AQUA']
        prefixes = {'TERRA':'MOD','AQUA':'MYD'}
        print ''
        print 'Use both TERRA and AQUA Satellites'
        print ''

    #platforms = ['TERRA','AQUA']
    #prefixes = {'TERRA':'MOD','AQUA':'MYD'}

    # ----- End modification


    granules = []
    # get geodata files and extract relevant info
    for platform in platforms:
        print('Searching imagery from ' +  platform + '...')
        for day in days:
            print('Acquiring metadata...')
            filename = getgeodatatxt(platform,prefixes[platform],day)
            print('Searching metadata to identify granules...')
            granules = parsegeodata(granules,filename,latbnds,lonbnds)
    # remove night passes
    print('Ignoring imagery aquired during non-daylight hours')
    granules = [x for x in granules if x.DayNight=='D']
    return granules

# retrieval of geodata text file
def getgeodatatxt(platform,prefix,day):
  filename = prefix + '03_' +  day.strftime('%Y-%m-%d') + '.txt'
  #connecting ftp
  print('Connecting to ladsftp.nascom.nasa.gov')
  ftp = FTP('ladsftp.nascom.nasa.gov')
  ftp.login()
  # changing directory
  ftp.cwd('/'.join(['geoMeta','6',platform,day.strftime('%Y')]))
  print('Downloading: ' + filename)
  with open(filename,'wb') as f:
    ftp.retrbinary('RETR ' + filename, f.write)
  print('Closing connection')
  ftp.close()
  return filename

# parse files and return granules of interest
def parsegeodata(granules,geofile,latbnds,lonbnds):
```

```
    eventLat = [latbnds[0],latbnds[1],latbnds[1],latbnds[0]]
    eventLon = [lonbnds[0],lonbnds[0],lonbnds[1],lonbnds[1]]
    eventPoly = Polygon(zip(eventLon,eventLat))
    with open(geofile,'r') as f:
        for line in f:
            if line[0]=='#' and len(line)>0:
                continue
            line = line.split(',')
            lon = map(float,line[9:13])
            lat = map(float,line[13:17])
            granulePoly = Polygon(zip(lon,lat))
            eastbound = float(line[5])
            westbound = float(line[8])
            # record if granule intersects aoi (and doesn't cross date line)
            if eventPoly.intersects(granulePoly) and eastbound>westbound:
                print('>' + line[0])
                starttime = datetime.datetime.strptime(line[1],'%Y-%m-%d
%H:%M')
                granules.append(granule(line[0],'',starttime,line[2],\
                    float(line[3]),line[4],lon,lat))
    print('Removing ' + geofile + ' from disk')
    os.remove(geofile)
    return granules

# download modis files
def downloadmodis(file03,modTime,archive):
    #connecting ftp
    print('Connecting to ladsftp.nascom.nasa.gov')
    ftp = FTP('ladsftp.nascom.nasa.gov')
    ftp.login()
    # download 03 product
    print('Downloading geolocation info...')
    path03 = '/'.join(['allData',archive,file03[0:5],modTime.strftime('%Y'),\
        modTime.strftime('%j')])
    ftp.cwd(path03)
    with open(file03,'wb') as f:
        ftp.retrbinary('RETR ' + file03, f.write)
    # download 021KM product (find file in directory that matches time)
    print('Downloading imagery...')
    path02 = '/'.join(['/allData',archive,file03[0:3]+'021KM',\
        modTime.strftime('%Y'),modTime.strftime('%j')])
    ftp.cwd(path02)
    filelist = ftp.nlst()

    '''
    # modification
    print filelist
    print len(filelist)
    print ''
    print file03
    print len(file03)
    print file03.split('.')[2]
    # end modification
    '''

    file02 = [x for x in filelist if x.split('.')[2] ==
file03.split('.')[2]][0]
```

```
    with open(file02,'wb') as f:
        ftp.retrbinary('RETR ' + file02, f.write)
    print('Disconnecting')
    ftp.close()
    return file02

# set flags to appropriate values or nan
def removeflags(A):
    A = A.astype('float64')
    A[A==65533] = 32767 #saturated
    A[A==65532] = 0 #zero point
    A[A==65530] = 0 #below valid range
    A[A==65529] = 32787 #above valid range
    A[A>32787] = np.nan #all other flags
    return A

# convert radiance to brightness temperatures
def rad2tb(rad,wavelength):
    planck = 6.626176E-34 # J sec
    boltz  = 1.380662E-23 # J/K
    cLight = 2.99792458E+8 # m/sec
    c1 = 2 * planck * cLight**2
    c2 = planck * cLight / boltz
    V = 1/wavelength # 1/m
    R = rad*1e6 # 1e6 micron/m
    TB = c2*V/np.log(c1*V**5/R + 1.0)
    return TB

# scale data gamma correction
def scaledata(vals,minin,maxin,minout,maxout,gamma):
    vals = vals - minin
    vals = vals/(maxin-minin)
    vals[np.isnan(vals)] = 0
    vals[vals<0] = 0
    vals[vals>1] = 1
    if not gamma == 1.0:
        vals = vals**gamma
    if not (minout == 0 and maxout == 1):
        vals = vals*(maxout-minout) + minout
    return vals

#
def readdata(hdfDataFile,hdfGeoFile,latbnds,lonbnds,dlon,dlat):
    # read lat/lon grid
    print('Reading geolocation data...')
    hdfGeo = SD.SD(hdfGeoFile)
    sds = hdfGeo.select('Longitude')
    lon = sds.get()
    sds = hdfGeo.select('Latitude')
    lat = sds.get()

    # read land/sea mask
    print('Reading land/sea mask..')
    sds = hdfGeo.select('Land/SeaMask')
    landSeaMask = sds.get()

    # read data - emissive
```

```
    print('Reading emissive data...')
    i29 = 8
    i31 = 10
    i32 = 11
    hdf = SD.SD(hdfDataFile)
    sds = hdf.select('EV_1KM_Emissive')
    attr = sds.attributes()
    dimdata = sds.info()[2]
    band29 = sds.get(start =(i29,0,0),count=(1,dimdata[1],dimdata[2]))
    band31 = sds.get(start =(i31,0,0),count=(1,dimdata[1],dimdata[2]))
    band32 = sds.get(start =(i32,0,0),count=(1,dimdata[1],dimdata[2]))
    band29 = np.squeeze(band29)
    band31 = np.squeeze(band31)
    band32 = np.squeeze(band32)
    offsetsEmissive = attr['radiance_offsets']
    scalesEmissive  = attr['radiance_scales']

    # read data - band 26
    ('Reading band 26...')
    sds = hdf.select('EV_Band26')
    attr = sds.attributes()
    band26 = sds.get()
    offsetBand26 = attr['reflectance_offsets']
    scaleBand26  = attr['reflectance_scales']

    # read data - reflective (250m aggregated)
    print('Reading reflective data...')
    i1 = 0
    i2 = 1
    sds = hdf.select('EV_250_Aggr1km_RefSB')
    attr = sds.attributes()
    band1 = sds.get(start =(i1,0,0),count=(1,dimdata[1],dimdata[2]))
    band2 = sds.get(start =(i2,0,0),count=(1,dimdata[1],dimdata[2]))
    band1 = np.squeeze(band1)
    band2 = np.squeeze(band2)
    offsetsRef250 = attr['reflectance_offsets']
    scalesRef250  = attr['reflectance_scales']

    # read data - reflective (500m aggregated)
    i3 = 0
    i4 = 1
    sds = hdf.select('EV_500_Aggr1km_RefSB')
    attr = sds.attributes()
    band3 = sds.get(start =(i3,0,0),count=(1,dimdata[1],dimdata[2]))
    band4 = sds.get(start =(i4,0,0),count=(1,dimdata[1],dimdata[2]))
    band3 = np.squeeze(band3)
    band4 = np.squeeze(band4)
    offsetsRef500 = attr['reflectance_offsets']
    scalesRef500  = attr['reflectance_scales']

    # trim data to AOI
    print('Trimming data to area of interest')
    iAOI = np.logical_and(np.logical_and(lon>=lonbnds[0]-
dlon,lon<=lonbnds[1]+dlon),\
            np.logical_and(lat>=latbnds[0]-dlat,lat<=latbnds[1]+dlat))
    lat = lat[iAOI]
    lon = lon[iAOI]
```

```
    band29 = band29[iAOI]
    band31 = band31[iAOI]
    band32 = band32[iAOI]
    band1  = band1[iAOI]
    band2  = band2[iAOI]
    band3  = band3[iAOI]
    band4  = band4[iAOI]
    band26 = band26[iAOI]
    landSeaMask = landSeaMask[iAOI]

    return lon,lat,landSeaMask,band1,band2,band3,band4,band26,band29,band31,\
        band32,offsetsEmissive,scalesEmissive,offsetsRef250,scalesRef250,\
        offsetsRef500,scalesRef500,offsetBand26,scaleBand26,\
        i1,i2,i3,i4,i29,i31,i32

# convert data from digital numbers to physical values of interest
def convertdata(band1,band2,band3,band4,band26,band29,band31,band32,\
        offsetsEmissive,scalesEmissive,offsetsRef250,scalesRef250,\
        offsetsRef500,scalesRef500,offsetBand26,scaleBand26,\
        i1,i2,i3,i4,i29,i31,i32):
    # set missing values to min/max as appropriate or mark as nan
    print('Cleaning up missing values')
    band29 = removeflags(band29)
    band31 = removeflags(band31)
    band32 = removeflags(band32)
    band1  = removeflags(band1)
    band2  = removeflags(band2)
    band3  = removeflags(band3)
    band4  = removeflags(band4)
    band26 = removeflags(band26)

    # convert data to radiance/reflectance
    print('Converting digital numbers to radiance/reflectance')
    band29 = (band29-offsetsEmissive[i29])*scalesEmissive[i29]
    band31 = (band31-offsetsEmissive[i31])*scalesEmissive[i31]
    band32 = (band32-offsetsEmissive[i32])*scalesEmissive[i32]
    band1  = (band1-offsetsRef250[i1])*scalesRef250[i1]
    band2  = (band2-offsetsRef250[i2])*scalesRef250[i2]
    band3  = (band3-offsetsRef500[i3])*scalesRef500[i3]
    band4  = (band4-offsetsRef500[i4])*scalesRef500[i4]
    band26 = (band26-offsetBand26)*scaleBand26

    # convert emissive data to brightness temperatures
    print('Converting radiance to brightness temperature for emissive bands')
    band29 = rad2tb(band29,8.55E-6)
    band31 = rad2tb(band31,11.03E-6)
    band32 = rad2tb(band32,12.02E-6)

    return band1,band2,band3,band4,band26,band29,band31,band32

def readandpreprocessdata(hdfDataFile,hdfGeoFile,latbnds,lonbnds,dlon,dlat):

    lon,lat,landSeaMask,band1,band2,band3,band4,band26,band29,band31,band32,\
        offsetsEmissive,scalesEmissive,offsetsRef250,scalesRef250,\
        offsetsRef500,scalesRef500,offsetBand26,scaleBand26,\
        i1,i2,i3,i4,i29,i31,i32 = \
            readdata(hdfDataFile,hdfGeoFile,latbnds,lonbnds,dlon,dlat)
```

```
    band1,band2,band3,band4,band26,band29,band31,band32 = \
        convertdata(band1,band2,band3,band4,band26,band29,band31,band32,\
            offsetsEmissive,scalesEmissive,offsetsRef250,scalesRef250,\
            offsetsRef500,scalesRef500,offsetBand26,scaleBand26,\
            i1,i2,i3,i4,i29,i31,i32)

    return lon,lat,band1,band2,band3,band4,band26,band29,band31,band32,
landSeaMask

# write data to geotiff
def writeGeoTIFF(R,G,B,lonbnds,latbnds,dlon,dlat,outfile):
    # write data to geotiff
    driver = gdal.GetDriverByName('GTiff')
    gtiff = driver.Create(outfile,R.shape[0],R.shape[1],3,gdal.GDT_Byte)
    gtiff.GetRasterBand(1).WriteArray(np.transpose(R))
    gtiff.GetRasterBand(2).WriteArray(np.transpose(G))
    gtiff.GetRasterBand(3).WriteArray(np.transpose(B))

    # set projection information
    gtiff.SetGeoTransform([lonbnds[0],dlon,0,latbnds[1],0,-dlat])
    srs = osr.SpatialReference()
    srs.SetWellKnownGeogCS('WGS84')
    gtiff.SetProjection(srs.ExportToWkt())

# true color composite
def truecolor(band1,band3,band4):
    Rmin = 0.1
    Rmax = 0.67
    Rgamma = 0.5
    R = scaledata(band1,Rmin,Rmax,0,255,Rgamma)
    Gmin = 0.1
    Gmax = 0.67
    Ggamma = 0.5
    G = scaledata(band4,Gmin,Gmax,0,255,Ggamma)
    Bmin = 0.1
    Bmax = 0.67
    Bgamma = 0.5
    B = scaledata(band3,Bmin,Bmax,0,255,Bgamma)
    return R,G,B

# EU-METSAT False color algorithm
def falsecolorEUMETSAT(band29,band31,band32):
    Rmin = -4
    Rmax = 2
    R = scaledata(band32-band31,Rmin,Rmax,0,255,1.0)
    Gmin = 0
    Gmax = 15
    Ggamma = 2.5
    G = scaledata(band31-band29,Gmin,Gmax,0,255,Ggamma)
    Bmin = -12
    Bmax = 16
    B = scaledata(band31-273.15,Bmin,Bmax,0,255,1.0)
    return R,G,B

# NRL False color algorithm (Miller, 2003; GRL)
def falsecolorNRL(band1,band2,band3,band4,band26,band31,band32,landSeaMask):
```

```
# compute NRL false color dust composite
Gmin = -1.45
Gmax = 0
G = scaledata(np.log10(band3),Gmin,Gmax,0,255,1)
Bmin = -1.45
Bmax = 0
B = scaledata(np.log10(band4),Bmin,Bmax,0,255,1)
Rmin = -0.4
Rmax = 0.15
normdiff = (band2-band3)/(band2+band3)
R = scaledata(np.log10(normdiff),Rmin,Rmax,0,255,1)

# set to < 6 to use over water algorithm
# set to < 221 to use over land algorithm everywhere
iland = landSeaMask < 221

# compute NRL red values over land
L1min = -2
L1max = 2
L1 = scaledata( band32[iland]-band31[iland],L1min,L1max,0,1,1)
L2max = np.nanmax(band31)
if L2max < 301:
L2min = L2max - 21
else:
L2min = (L2max-273)/4+273
L2 = scaledata( band31[iland],L2min,L2max,0,1,1)
L3min = -1.5
L3max = 0.25
L3 = scaledata( 2*band1[iland]-band3[iland]-band4[iland]-
L2,L3min,L3max,0,1,1)
L4 = band26[iland] > 0.05
Rmin = 1.3
Rmax = 2.7
R[iland] = scaledata(L1+L3-L4+1-L2,Rmin,Rmax,0,255,1)
return R, G, B




def createfalsecolor(latbnds,lonbnds,hdfGeoFiles,hdfDataFiles,start-
time,aoitag):
    # resampling resolution
    dlat = 0.01
    dlon = 0.01


    # read data from hdf file and convert to desired units
    for i in range(len(hdfGeoFiles)):
        if i == 0:

lon,lat,band1,band2,band3,band4,band26,band29,band31,band32,landSeaMask = \
                readandpreprocessdata(hdfDataFiles[i],hdfGeoFiles[i],lat-
bnds,lonbnds,dlon,dlat)
        else:

loni,lati,band1i,band2i,band3i,band4i,band26i,band29i,band31i,band32i,landSea
Maski = \
```

```
            readandpreprocessdata(hdfDataFiles[i],hdfGeoFiles[i],lat-
bnds,lonbnds,dlon,dlat)
            lon = np.append(lon,loni)
            lat = np.append(lat,lati)
            band1 = np.append(band1,band1i)
            band2 = np.append(band2,band2i)
            band3 = np.append(band3,band3i)
            band4 = np.append(band4,band4i)
            band26 = np.append(band26,band26i)
            band29 = np.append(band29,band29i)
            band31 = np.append(band31,band31i)
            band32 = np.append(band32,band32i)
            landSeaMask = np.append(landSeaMask,landSeaMaski)



    #delete hdf files
    print('Removing raw imagery and geolocation data from disk')
    for f in hdfDataFiles:
        os.remove(f)
    for f in hdfGeoFiles:
        os.remove(f)

    # true color image
    print('Creating true color composite')
    Rtrue,Gtrue,Btrue = truecolor(band1,band3,band4)

    # compute EU-METSAT false color dust composite
    print('Creating EU-METSAT false color dust composite')
    Reumetsat,Geumetsat,Beumetsat = falsecolorEUMETSAT(band29,band31,band32)



    # compute NRL false color dust composite
    print('Creating NRL false color dust composite')
    Rnrl,Gnrl,Bnrl = \
        falsecolorNRL(band1,band2,band3,band4,band26,band31,band32,landSea-
Mask)


    # reshape/concat lat/lon data for interpolation (not a regular grid)
    latlon = np.concatenate((lon.reshape((-1,1)),lat.reshape((-1,1))),axis=1)

    # output coordinates
    longrid, latgrid = np.mgrid[lonbnds[0]:lonbnds[1]:dlon,latbnds[1]:lat-
bnds[0]:-dlat]


    # resample
    print('Resampling true color composite...')
    RtrueGrid   = griddata(latlon,Rtrue,(longrid, latgrid), method='linear')
    GtrueGrid   = griddata(latlon,Gtrue,(longrid, latgrid), method='linear')
    BtrueGrid   = griddata(latlon,Btrue,(longrid, latgrid), method='linear')
    print('Resampling EU-METSAT false color dust composite...')
    ReumetsatGrid = griddata(latlon,Reumetsat,(longrid, latgrid),
method='linear')
```

```
    GeumetsatGrid = griddata(latlon,Geumetsat,(longrid, latgrid),
method='linear')
    BeumetsatGrid = griddata(latlon,Beumetsat,(longrid, latgrid),
method='linear')


    print('Resampling NRL false color dust composite...')
    RnrlGrid     = griddata(latlon,Rnrl,(longrid, latgrid), method='linear')
    GnrlGrid     = griddata(latlon,Gnrl,(longrid, latgrid), method='linear')
    BnrlGrid     = griddata(latlon,Bnrl,(longrid, latgrid), method='linear')



    # write data to GeoTIFF
    print('Saving output to GeoTIFF...')
    outfilebase = '.'.join(['./' + aoitag,starttime.strftime('%Y%m%d'),\
        starttime.strftime('%H%M'),'MODIS'])
    writeGeoTIFF(ReumetsatGrid,GeumetsatGrid,BeumetsatGrid,\
                lonbnds,latbnds,dlon,dlat,'.'.join([outfilebase,'EU-
METSAT','tif']))


    writeGeoTIFF(RnrlGrid,GnrlGrid,BnrlGrid,\
                lonbnds,latbnds,dlon,dlat,'.'.join([out-
filebase,'NRL','tif']))


    writeGeoTIFF(RtrueGrid,GtrueGrid,BtrueGrid,\
                lonbnds,latbnds,dlon,dlat,'.'.join([outfilebase,'TrueCol-
or','tif']))

# create geotiffs
def processdata(granules,latbnds,lonbnds,aoi):
    # (granules from the same platform during the same orbit are combined)
    geoFiles  = [granules[0].File03]
    dataFiles = [granules[0].File02]
    startTime = granules[0].StartTime
    searchTime = granules[0].StartTime
    for i in range(1,len(granules)):
        searchTime = searchTime + datetime.timedelta(minutes=5)
        # if next granule is in continuous set, add on
        if granules[i].StartTime == searchTime:
            geoFiles.append(granules[i].File03)
            dataFiles.append(granules[i].File02)
        # otherwise, create geotiffs and start new set
        else:
            if len(geoFiles) == 1:
                print('Processing granule ' + str(i) + ' of ' \
                    + str(len(granules)) + '...')
            else:
                print('Processing granules ' + str(i-len(geoFiles)+1) + \
                    ' through ' + str(i) + 'of' + str(len(granules)) + '...')
            createfalsecolor(latbnds,lonbnds,geoFiles,dataFiles,start-
Time,aoi)
            geoFiles   = [granules[i].File03]
            dataFiles  = [granules[i].File02]
            startTime  = granules[i].StartTime
```

```
            searchTime = granules[i].StartTime
    if len(geoFiles) == 1:
        print('Processing granule ' + str(i+1) + ' of ' \
            + str(len(granules)) + '...')
    else:
        print('Processing granules ' + str(i-len(geoFiles)+2) + \
            ' through ' + str(i+1) + ' of ' + str(len(granules)) + '...')
    createfalsecolor(latbnds,lonbnds,geoFiles,dataFiles,startTime,aoi)


# get user defined inputs
latbnds,lonbnds,date1,date2,aoi = getuserinput()




# find MODIS granules that cover AOI
print('Searching for MODIS granules that cover scene...')
granules = findgranules(latbnds,lonbnds,date1,date2)

# download files
print('Downloading data...')
for i in range(len(granules)):
    print('Granule ' + str(i+1) + ' of ' + str(len(granules)) + ':')
    granules[i].File02 = downloadmodis(granules[i].File03,\
                            granules[i].StartTime,granules[i].Archive)

# create GeoTIFFs
print('Processing data...')
processdata(granules,latbnds,lonbnds,aoi)

print('')
print('-------finished-------')
```

Appendix C: ArcMap Mapping Procedures

Create a folder entitled "Plume_Head_Mapping"; and within it, create "MODIS Imagery" and "Shapefiles" folders. Place all MODIS imagery deemed suitable for mapping into the MODIS Imagery folder. Open ArcMap and select the "Blank Map" template to open a new project file. To begin building the project file, add a basemap by clicking on the downward facing arrow on the right side of the "Add Data" button (Figure C-1). From the drop down menu, select "Add Basemap" and select "Imagery." High-resolution (1 m) satellite imagery of the globe will load into the project file and will be listed under the Table of Contents on the left-hand side of the screen (see Figure C-1). To add MODIS imagery, click on the Arc Catalog button (see Figure C-1). Right click "Folder Connections," select "Connect to Folder," navigate to the "Plume_Head_Mapping" folder, and press "OK." The "Plume_Head_Mapping" folder will appear under "Folder Connections" and can be expanded by clicking the small plus sign to the left of the folder name. Open the MODIS Imagery folder located within Plume_Head_Mapping, and all of the images within that folder will appear in the Arc Catalog (see Figure C-1). To add an image to the project file, simply click on it in the Arc Catalog and drag it over to the Table of Contents.

Mapping is conducted by creating and editing point shapefiles. Shapefiles are spatially oriented points, lines, or polygons that are representative of other data (e.g., features on a map). To create a new shapefile, open Arc Catalog; and navigate to the Shapefiles folder, right click on it, and select "New," then "Shapefile," and a "Create New Shapefile" window will appear. Enter a name, and select "Point" under "Feature Type." Under "Description," click "Edit," and a "Spatial Reference Properties" window will open. A commonly used coordinate system is WGS 1984, or the World Geodetic System 1984. To navigate to this system, click the "Geographic Coordinate System" folder, click the "World" folder, highlight "WGS 1984," and click "OK."

Figure C-1. A screenshot of an ArcMap project file. The *top* panel shows an enlarged version of the top left corner of the project file. Within the top panel, the *red oval* depicts the location of the Add Data button while the *blue oval* highlights the location of the Table of Contents. The *green oval* depicts the location of the Arc Catalog, which, when selected, opens a catalog on the right-hand side of the project file. The *pink oval* shows the location of the Editor Toolbar. The expanded Arc Catalog is shown in the *lower* panel, and the contents of the MODIS Imagery folder are enlarged.
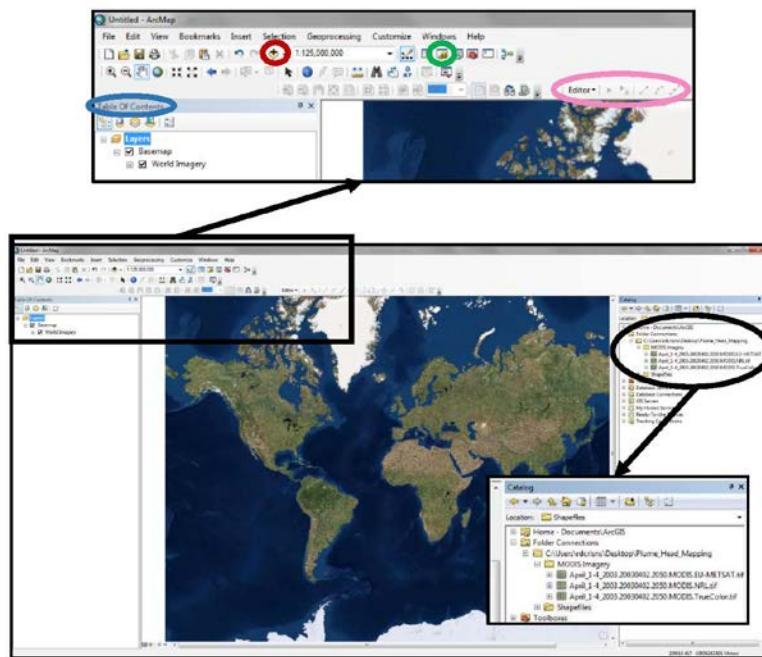
To map plume heads, click "Customize" at the top of the screen, select "Toolbars," and click on "Editor." The Editor Toolbar will appear in the project file (see Figure C-1). In the Editor Toolbar, click "Editor" and then "Start Editing." To the right of the screen, a "Create Features" window will appear. If this window is not visible, click on the pull-down menu on the Editor Toolbar, expand "Editing Windows," and select "Create Features." In the "Create Features" window, click on the name of the shapefile that will be edited to activate the tool that allows users to place new points. Once activated, a new point will be placed on the map anywhere a user clicks. If unwanted points are accidentally placed, remove them by selecting the black arrow in the Editor Toolbar, highlighting the unwanted point, and pressing delete. To save any edits, click "Editor" on the Editor Toolbar, and select "Save Edits."

1. REPORT DATE (DD-MM-YYYY) May 2017	**2. REPORT TYPE** Technical Report/Final	**3. DATES COVERED** (From - To)

4. TITLE AND SUBTITLE Subjective Mapping of Dust-Emission Sources by Using MODIS Imagery: Reproducibility Assessment	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Samantha N. Sinclair and Sandra L. Jones	**5d. PROJECT NUMBER**
	5e. TASK NUMBER ERDC 6.2
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center (ERDC) Cold Regions Research and Engineering Laboratory (CRREL) 72 Lyme Road Hanover, NH 03755-1290	**8. PERFORMING ORGANIZATION REPORT NUMBER** ERDC/CRREL TR-17-8

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Assistant Secretary of the Army for Acquisition, Logistics, and Technology 103 Army Pentagon Washington, DC 20314-1000	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

ERDC 6.2 Geospatial Research and Engineering (GRE) ARTEMIS STO-R DUST-CLOUD

14. ABSTRACT

Dust storms (5 to 100 km across) often originate from multiple dust-emission sources (1 to 10 km across). Remote-sensing-based dust-source identification is a challenge. A previous study developed a subjective approach for mapping dust sources by using enhanced MODIS satellite imagery; therefore, this study conducted mapping exercises to assess the reproducibility of this technique amongst multiple analysts and in different regions. Multiple staff members independently analyzed satellite imagery for mappable dust sources for Southwest Asia and the Southwest United States. Mapped points were considered reproducible if the location of dust emission plumes identified by all participants could be constrained to a 10 km buffer. Using this definition, point-source locations were 28% reproducible in Southwest Asia and 85% reproducible in the Southwest United States. Increasing the allotted buffer to 15 km, however, improved results to 71% and 100%, respectively. Mapped dust sources for Southwest Asia were compared to geomorphic landform maps. At the 1:750,000 map scale, points mapped by all analysts for a single dust plume tended to overlay one landform, while at the 1:100,000 map scale, points were strewn across several landforms. Results suggest that the methodology is reproducible for certain applications but that location-uncertainty tolerance affects perceived conclusions.

15. SUBJECT TERMS

Airborne dust, Dust detection, Dust emission, Dust storms, GIS, MODIS, Remote sensing, Remote-sensing images, Reproducibility

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	61	**19b. TELEPHONE NUMBER** (include area code)